# Model Checking for Fragments of Halpern and Shoham's Interval Temporal Logic Based on Track Representatives[☆]

Alberto Molinari[a], Angelo Montanari[a], Adriano Peron[b]

[a] *Department of Mathematics, Computer Science, and Physics, University of Udine, Italy*
[b] *Department of Electronic Engineering and IT, University of Napoli "Federico II", Italy*

**Abstract**

Model checking allows one to automatically verify a specification of the expected properties of a system against a formal model of its behaviour (generally, a Kripke structure). Point-based temporal logics, such as LTL, CTL, and CTL*, that describe how the system evolves state-by-state, are commonly used as specification languages. They proved themselves quite successful in a variety of application domains. However, properties constraining the temporal ordering of temporally extended events as well as properties involving temporal aggregations, which are inherently interval-based, can not be properly dealt with by them. Interval temporal logics (ITLs), that take intervals as their primitive temporal entities, turn out to be well-suited for the specification and verification of interval properties of computations (we interpret all the tracks of a Kripke structure as computation intervals).

In this paper, we study the model checking problem for some fragments of Halpern and Shoham's modal logic of time intervals (HS). HS features one modality for each possible ordering relation between pairs of intervals (the so-called Allen's relations). First, we describe an EXPSPACE model checking algorithm for the HS fragment of Allen's relations *meets*, *met-by*, *starts*, *started-by*, and *finishes*, which exploits the possibility of finding, for each track (of unbounded length), an equivalent bounded-length track representative. While checking a property, it only needs to consider tracks whose length does not exceed the given bound. Then, we prove the model checking problem for such a fragment to be PSPACE-hard. Finally, we identify other well-behaved HS fragments which are expressive enough to capture meaningful interval properties of systems, such as mutual exclusion, state reachability, and non-starvation, and whose computational complexity is less than or equal to that of LTL.

*Keywords:* Model checking, interval temporal logics, computational complexity
*2010 MSC:* 03B70, 68Q60

[☆] This paper is an extended and revised version of [22] and [21].

*Email addresses:* molinari.alberto@gmail.com (Alberto Molinari), angelo.montanari@uniud.it (Angelo Montanari), adrperon@unina.it (Adriano Peron)

## 1. Introduction

One of the most notable techniques for system verification is model checking, which allows one to verify the desired properties of a system against a model of its behaviour [9]. Properties are usually formalized by means of temporal logics, such as LTL and CTL, and systems are represented as labelled state-transition graphs (Kripke structures). Model checking algorithms perform, in a fully automatic way, an (implicit or explicit) exhaustive enumeration of all the states reachable by the system, and either terminate positively, proving that all properties are met, or produce a counterexample, witnessing that some behavior falsifies a property.

The model checking problem has systematically been investigated in the context of classical, point-based temporal logics, like LTL, CTL, and CTL$^*$, which predicate over single computation points/states, while it is still largely unexplored in the interval logic setting.

Interval temporal logics (ITLs) have been proposed as a formalism for temporal representation and reasoning more expressive than standard point-based ones [13, 34, 35]. They take intervals, instead of points, as their primitive temporal entities. Such a choice gives them the ability to cope with advanced temporal properties, such as actions with duration, accomplishments, and temporal aggregations, which can not be properly dealt with by standard, point-based temporal logics.

Expressiveness of ITLs makes them well suited for many applications in a variety of computer science fields, including artificial intelligence (reasoning about action and change, qualitative reasoning, planning, configuration and multi-agent systems, and computational linguistics), theoretical computer science (formal verification, synthesis), and databases (temporal and spatio-temporal databases) [2, 10, 18, 30, 8, 27, 26, 19, 11]. However, this great expressiveness is a double-edged sword: in most cases the satisfiability problem for ITLs turns out to be undecidable, and, in the few cases of decidable ITLs, the standard proof machinery, like Rabin's theorem, is usually not applicable.

The most prominent ITL is Halpern and Shoham's modal logic of time intervals (HS, for short) [13]. HS features one modality for each of the 13 possible ordering relations between pairs of intervals (the so-called Allen's relations [1]), apart from the equality relation. In [13], it has been shown that the satisfiability problem for HS interpreted over all relevant (classes of) linear orders is undecidable. Since then, a lot of work has been done on the satisfiability problem for HS fragments, which has shown that undecidability prevails over them (see [4] for an up-to-date account of undecidable fragments). However, meaningful exceptions exist, including the interval logic of temporal neighbourhood $\mathsf{A\overline{A}}$ and the interval logic of sub-intervals $\mathsf{D}$ [5, 6, 7, 25].

In this paper, we focus our attention on the model checking problem for HS, for which, as we said, little work has been done [24, 20, 15, 16, 17] (it is worth pointing out that, in contrast to the case of point-based, linear temporal logics, there is not an easy reduction from the model checking problem to validity/satisfiability for ITL).

**Related work.** In the classical formulation of the model checking problem [9], point-based temporal logics are used to analyze, for each path in a Kripke structure, how proposition letters labelling the states change from one state to the next one along the path. In interval-based model checking, in order to check interval properties of computations, one needs to collect information about states into computation stretches. This amounts to interpreting each finite path of a Kripke structure (a track) as an interval, and to suitably defining its labelling on the basis of the proposition letters that hold on the states composing it.

In [24], Montanari et al. give a first characterization of the model checking problem for full HS, interpreted over finite Kripke structures (under the homogeneity assumption [31], according to which a proposition letter holds on an interval if and only if it holds on all its sub-intervals). In that paper, the authors introduce the basic elements of the general picture, namely, the interpretation of HS formulas over (abstract) interval models, the mapping of finite Kripke structures into (abstract) interval models, the notion of track descriptor, and a small model theorem proving (with a non-elementary procedure) the decidability of the model checking problem for full HS against finite Kripke structures. Many of these notions will be recalled in the following section. In [20], Molinari et al. work out the model checking problem for full HS in all its details, and prove that it is EXPSPACE-hard, if a succinct encoding of formulas is allowed, and PSPACE-hard otherwise.

In [15, 16, 17], Lomuscio and Michaliszyn address the model checking problem for some fragments of HS extended with epistemic modalities. Their semantic assumptions differ from those made in [24], making it difficult to compare the outcomes of the two research directions. In both cases, formulas of interval temporal logic are evaluated over finite paths/tracks obtained from the unravelling of a finite Kripke structure. However, while in [24] a proposition letter holds over an interval (track) if and only if it holds over all its states (homogeneity assumption), in [15, 16] truth of proposition letters on a track/interval depends only on their values at its endpoints.

In [15], the authors focus their attention on the HS fragment $\mathsf{BED}$ of Allen's relations *started-by*, *finished-by*, and *contains* (since modality $\langle \mathrm{D} \rangle$ is definable in terms of modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$, $\mathsf{BED}$ is actually as expressive as $\mathsf{BE}$), extended with epistemic modalities. They consider a restricted form of model checking, which verifies the given specification against a single (finite) initial computation interval. Their goal is indeed to reason about a given computation of a multi-agent system, rather than on all its admissible computations. They prove that the considered model checking problem is PSPACE-complete; moreover, they show that the same problem restricted to the pure temporal fragment $\mathsf{BED}$, that is, the one obtained by removing epistemic modalities, is in PTIME. These results do not come as a surprise as they trade expressiveness for efficiency: modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ allow one to access only sub-intervals of the initial one, whose number is quadratic in the length (number of states) of the initial interval.

In [16], they show that the picture drastically changes with other fragments of HS, that allow one to access infinitely many tracks/intervals. In particular, they prove that the model checking problem for the HS fragment $\mathsf{A\overline{B}L}$ of Allen's relations *meets*, *starts*, and *before* (since modality $\langle \mathrm{L} \rangle$ is definable in terms of modality $\langle \mathrm{A} \rangle$, $\mathsf{A\overline{B}L}$ is actually as expressive as $\mathsf{A\overline{B}}$), extended with epistemic modalities, is decidable with a non-elementary upper bound. Note that, thanks to modalities $\langle \mathrm{A} \rangle$ and $\langle \overline{\mathrm{B}} \rangle$, formulas of $\mathsf{A\overline{B}L}$ can possibly refer to infinitely many (future) tracks/intervals.

Finally, in [17], Lomuscio and Michaliszyn show how to use regular expressions in order to specify the way in which tracks/intervals of a Kripke structure get labelled. Such an extension leads to a significant increase in expressiveness, as the labelling of an interval is no more determined by that of its endpoints, but it depends on the ordered sequence of states the interval consists of. They also prove that there is not a corresponding increase in computational complexity, as the complexity bounds given in [15, 16] still hold with the new semantics: the model checking problem for $\mathsf{BED}$ is still in PSPACE, and it is non-elementarily decidable for $\mathsf{A\overline{B}L}$.

**Main contributions.** In this paper, we elaborate on the approach to ITL model checking outlined in [24] and we propose an original solution to the problem for some relevant HS fragments based on the notion of track representative. We first prove that the model checking problem for two large HS fragments, namely, the fragment $\mathsf{A\overline{A}B\overline{B}E}$ (resp., $\mathsf{A\overline{A}E\overline{B}E}$) of Allen's relations *meets*, *met-by*,

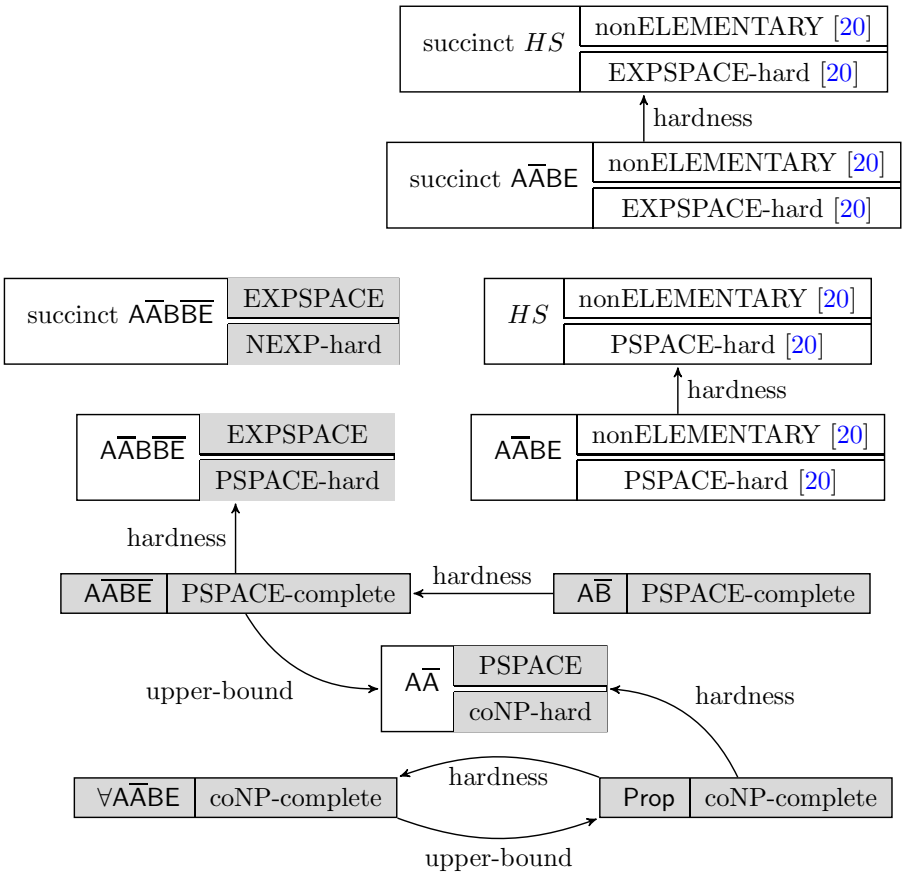



Figure 1: Complexity of model checking for HS fragments.

*started-by* (resp., *finished-by*), *starts* and *finishes*, is in EXPSPACE. Moreover, we show that it is PSPACE-hard (NEXP-hard, if a succinct encoding of formulas is used). Then, we identify some well-behaved HS fragments, which are still expressive enough to capture meaningful interval properties of state-transition systems, such as mutual exclusion, state reachability, and non-starvation, whose model checking problem exhibits a considerably lower computational complexity, notably, (*i*) the fragment $\mathsf{A\overline{A}\overline{B}E}$, whose model checking problem is PSPACE-complete, and (*ii*) the fragment $\forall\mathsf{A\overline{A}BE}$, including formulas of $\mathsf{A\overline{A}BE}$ where only universal modalities are allowed and negation can be applied to propositional formulas only, whose model checking problem is coNP-complete.

In Figure 1, we summarize known (white boxes) and new (grey boxes) results about complexity of model checking for HS fragments.

The main technical contributions of the paper can be summarized as follows.

- *Track descriptors.* We start with some background knowledge about HS and Kripke structures, and then we show how the latter can be mapped into interval-based structures, called *abstract interval models*, over which HS formulas are evaluated. Each track in a Kripke structure is interpreted as an interval, which becomes an (atomic) object of the domain of an abstract interval model. The labeling of an interval is defined on the basis of the states that compose it, according to the homogeneity assumption [31]. Then, we introduce *track descriptors* [24]. A track descriptor is a tree-like structure providing information about a possibly

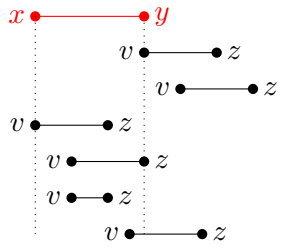
infinite set of tracks (the number of admissible track descriptors for a given Kripke structure is finite). Being associated with the same descriptor is indeed a sufficient condition for two tracks to be indistinguishable with respect to satisfiability of $\mathsf{A\overline{A}B\overline{B}\overline{E}}$ formulas, provided that the nesting depth of $\langle \mathrm{B} \rangle$ modality is less than or equal to the depth of the descriptor itself. Finally, we introduce the key notions of *descriptor sequence for a track* and *cluster*, and the relation of *descriptor element indistinguishability*, which allow us to determine when two prefixes of some track are associated with the same descriptor, avoiding the expensive operation of explicitly constructing track descriptors.

- *A small model theorem.* The main result of the paper is a small model theorem, showing that we can restrict the verification of an $\mathsf{A\overline{A}B\overline{B}\overline{E}}$ formula to a finite number of bounded-length *track representatives*. A track representative is a track that can be analyzed in place of all—possibly infinitely many—tracks associated with its descriptor. We use track representatives to devise an EXPSPACE model checking algorithm for $\mathsf{A\overline{A}B\overline{B}\overline{E}}$. Descriptor element indistinguishability plays a fundamental role in the proof of the bound to the maximum length of representatives, and it allows us to show the completeness of the algorithm, which considers all the possible representatives. In addition, we prove that the model checking problem for $\mathsf{A\overline{A}B\overline{B}\overline{E}}$ is PSPACE-hard, NEXP-hard if a *succinct encoding* of formulas is used (it is worth noticing that the proposed algorithm requires exponential working space also in the latter case).

- *Well-behaved HS fragments.* We first show that the proposed model checking algorithm can verify formulas with a constant nesting depth of $\langle \mathrm{B} \rangle$ modality by using polynomial working space. This allows us to conclude that the model checking problem for $\mathsf{A\overline{A}B\overline{E}}$ formulas (which lack modality $\langle \mathrm{B} \rangle$) is in PSPACE. Then, we prove that the model checking problem for $\mathsf{A\overline{B}}$ is PSPACE-hard. PSPACE-completeness of $\mathsf{A\overline{A}B\overline{E}}$ (and $\mathsf{A\overline{B}}$) immediately follows. Next, we deal with the fragment $\forall\mathsf{A\overline{A}BE}$. We first provide a coNP model checking algorithm for $\forall\mathsf{A\overline{A}BE}$, and then we show that model checking for the pure propositional fragment $\mathsf{Prop}$ is coNP-hard. The two results together allow us to conclude that the model checking problem for both $\mathsf{Prop}$ and $\forall\mathsf{A\overline{A}BE}$ is coNP-complete. In addition, upper and lower bounds to the complexity of the problem for $\mathsf{A\overline{A}}$ (the logic of temporal neighbourhood) directly follow: since $\mathsf{A\overline{A}}$ is a fragment of $\mathsf{A\overline{A}B\overline{E}}$ and $\mathsf{Prop}$ is a fragment of $\mathsf{A\overline{A}}$, complexity of model checking for $\mathsf{A\overline{A}}$ is in between coNP and PSPACE.

**Organization of the paper.** In Section 2, we provide some background knowledge. Then, in Section 3, we introduce track descriptors [24] and, in Section 4, we formally define the key relation of indistinguishability over descriptor elements. In Section 5, we describe an EXPSPACE model checking algorithm for $\mathsf{A\overline{A}B\overline{B}\overline{E}}$ based on track representatives. We also show how to obtain a PSPACE model checking algorithm for $\mathsf{A\overline{A}B\overline{E}}$ by suitably tailoring the one for $\mathsf{A\overline{A}B\overline{B}\overline{E}}$. In Section 6, we prove that model checking for $\mathsf{A\overline{A}B\overline{E}}$ is PSPACE-hard; PSPACE-completeness immediately follows. Moreover, we get for free a lower bound to the complexity of the model checking problem for $\mathsf{A\overline{A}B\overline{B}\overline{E}}$, which turns out to be PSPACE-hard (in the appendix, we show that the problem is NEXP-hard if a succinct encoding of formulas is used). Finally, in Section 7 we provide a coNP model checking algorithm for $\forall\mathsf{A\overline{A}BE}$ and then we show that the problem is actually coNP-complete. Conclusions give a short assessment of the work done and describe future research directions.

Table 1: Allen's interval relations and corresponding HS modalities.

| Allen's relation | HS | Definition w.r.t. interval structures | Example |
|---|---|---|---|
| *meets* | $\langle \mathrm{A} \rangle$ | $[x,y]\mathcal{R}_A[v,z] \iff y = v$ |  |
| *before* | $\langle \mathrm{L} \rangle$ | $[x,y]\mathcal{R}_L[v,z] \iff y < v$ | |
| *started-by* | $\langle \mathrm{B} \rangle$ | $[x,y]\mathcal{R}_B[v,z] \iff x = v \wedge z < y$ | |
| *finished-by* | $\langle \mathrm{E} \rangle$ | $[x,y]\mathcal{R}_E[v,z] \iff y = z \wedge x < v$ | |
| *contains* | $\langle \mathrm{D} \rangle$ | $[x,y]\mathcal{R}_D[v,z] \iff x < v \wedge z < y$ | |
| *overlaps* | $\langle \mathrm{O} \rangle$ | $[x,y]\mathcal{R}_O[v,z] \iff x < v < y < z$ | |



# 2. Preliminaries

## 2.1. The interval temporal logic HS

Interval-based approaches to temporal representation and reasoning have been successfully pursued in computer science and artificial intelligence. An interval algebra to reason about intervals and their relative order was first proposed by Allen [1]. Then, a systematic logical study of ITLs was done by Halpern and Shoham, who introduced the logic HS featuring one modality for each Allen interval relation [13], except for equality. Table 1 depicts 6 of the 13 Allen's relations together with the corresponding HS (existential) modalities. The other 7 are equality and the 6 inverse relations (given a binary relation $\mathcal{R}$, the inverse relation $\overline{\mathcal{R}}$ is such that $b\overline{\mathcal{R}}a$ if and only if $a\mathcal{R}b$).

The language of HS features a set of proposition letters $\mathcal{AP}$, the Boolean connectives $\neg$ and $\wedge$, and a temporal modality for each of the (non trivial) Allen's relations, namely, $\langle \mathrm{A} \rangle$, $\langle \mathrm{L} \rangle$, $\langle \mathrm{B} \rangle$, $\langle \mathrm{E} \rangle$, $\langle \mathrm{D} \rangle$, $\langle \mathrm{O} \rangle$, $\langle \overline{\mathrm{A}} \rangle$, $\langle \overline{\mathrm{L}} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, $\langle \overline{\mathrm{E}} \rangle$, $\langle \overline{\mathrm{D}} \rangle$ and $\langle \overline{\mathrm{O}} \rangle$. HS formulas are defined by the following grammar:

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \langle X \rangle \psi \mid \langle \overline{X} \rangle \psi, \quad \text{with } p \in \mathcal{AP},\ X \in \{A, L, B, E, D, O\}.$$

We will make use of the standard abbreviations of propositional logic, e.g., we will write $\psi \vee \phi$ for $\neg(\neg\psi \wedge \neg\phi)$, $\psi \rightarrow \phi$ for $\neg\psi \vee \phi$, and $\psi \leftrightarrow \phi$ for $(\psi \rightarrow \phi) \wedge (\phi \rightarrow \psi)$. Moreover, for all $X$, dual universal modalities $[X]\psi$ and $[\overline{X}]\psi$ are defined as $\neg\langle X \rangle \neg\psi$ and $\neg\langle \overline{X} \rangle \neg\psi$, respectively.

We will assume the *strict semantics* of HS: only intervals consisting of at least two points are allowed. Under that assumption, HS modalities are *mutually exclusive* and *jointly exhaustive*, that is, exactly one of them holds between any two intervals. However, the strict semantics can easily be "relaxed" to include point intervals, and all results we are going to prove hold for the non-strict HS semantics as well. All HS modalities can be expressed in terms of $\langle \mathrm{A} \rangle$, $\langle \mathrm{B} \rangle$, and $\langle \mathrm{E} \rangle$, and the inverse modalities $\langle \overline{\mathrm{A}} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, and $\langle \overline{\mathrm{E}} \rangle$, as follows:

$$\begin{array}{cc} \langle \mathrm{L} \rangle\, \psi \equiv \langle \mathrm{A} \rangle \langle \mathrm{A} \rangle\, \psi & \langle \overline{\mathrm{L}} \rangle\, \psi \equiv \langle \overline{\mathrm{A}} \rangle \langle \overline{\mathrm{A}} \rangle\, \psi \\ \langle \mathrm{D} \rangle\, \psi \equiv \langle \mathrm{B} \rangle \langle \mathrm{E} \rangle\, \psi \equiv \langle \mathrm{E} \rangle \langle \mathrm{B} \rangle\, \psi & \langle \overline{\mathrm{D}} \rangle\, \psi \equiv \langle \overline{\mathrm{B}} \rangle \langle \overline{\mathrm{E}} \rangle\, \psi \equiv \langle \overline{\mathrm{E}} \rangle \langle \overline{\mathrm{B}} \rangle\, \psi \\ \langle \mathrm{O} \rangle\, \psi \equiv \langle \mathrm{E} \rangle \langle \overline{\mathrm{B}} \rangle\, \psi & \langle \overline{\mathrm{O}} \rangle\, \psi \equiv \langle \mathrm{B} \rangle \langle \overline{\mathrm{E}} \rangle\, \psi \end{array} .$$

We denote by $\mathsf{X_1} \cdots \mathsf{X_n}$ the fragment of HS that features modalities $\langle X_1 \rangle, \cdots, \langle X_n \rangle$ only.

HS can be viewed as a multi-modal logic with the 6 primitive modalities $\langle \mathrm{A} \rangle$, $\langle \mathrm{B} \rangle$, $\langle \mathrm{E} \rangle$, $\langle \overline{\mathrm{A}} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, and $\langle \overline{\mathrm{E}} \rangle$. Accordingly, HS semantics can be defined over a multi-modal Kripke structure, called here an *abstract interval model*, in which (strict) intervals are treated as atomic objects and Allen's relations as simple binary relations between pairs of them.

**Definition 1** ([20]). *An* abstract interval model *is a tuple* $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$*, where* $\mathcal{AP}$ *is a finite set of proposition letters,* $\mathbb{I}$ *is a possibly infinite set of atomic objects (worlds),* $A_{\mathbb{I}}$*,* $B_{\mathbb{I}}$*, and* $E_{\mathbb{I}}$ *are three binary relations over* $\mathbb{I}$*, and* $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ *is a (total) labeling function which assigns a set of proposition letters to each world.*

Intuitively, in the interval setting, $\mathbb{I}$ is a set of intervals, $A_{\mathbb{I}}$, $B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as Allen's interval relations $A$ (*meets*), $B$ (*started-by*), and $E$ (*finished-by*), respectively, and $\sigma$ assigns to each interval the set of proposition letters that hold over it.

Given an abstract interval model $\mathcal{A} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$ and an interval $I \in \mathbb{I}$, truth of an HS formula over $I$ is defined by structural induction on the formula as follows:

- $\mathcal{A}, I \models p$ if and only if $p \in \sigma(I)$, for any proposition letter $p \in \mathcal{AP}$;
- $\mathcal{A}, I \models \neg\psi$ if and only if it is not true that $\mathcal{A}, I \models \psi$ (also denoted as $\mathcal{A}, I \not\models \psi$);
- $\mathcal{A}, I \models \psi \wedge \phi$ if and only if $\mathcal{A}, I \models \psi$ and $\mathcal{A}, I \models \phi$;
- $\mathcal{A}, I \models \langle X \rangle \psi$, for $X \in \{A, B, E\}$, if and only if there exists $J \in \mathbb{I}$ such that $I \, X_{\mathbb{I}} \, J$ and $\mathcal{A}, J \models \psi$;
- $\mathcal{A}, I \models \langle \overline{X} \rangle \psi$, for $\overline{X} \in \{\overline{A}, \overline{B}, \overline{E}\}$, if and only if there exists $J \in \mathbb{I}$ such that $J \, X_{\mathbb{I}} \, I$ and $\mathcal{A}, J \models \psi$.

## 2.2. Kripke structures and abstract interval models

In this section, we define a mapping from Kripke structures to abstract interval models that makes it possible to specify properties of systems by means of HS formulas.

**Definition 2.** *A finite Kripke structure* $\mathcal{K}$ *is a tuple* $(\mathcal{AP}, W, \delta, \mu, w_0)$*, where* $\mathcal{AP}$ *is a set of proposition letters,* $W$ *is a finite set of states,* $\delta \subseteq W \times W$ *is a left-total relation between pairs of states,* $\mu : W \mapsto 2^{\mathcal{AP}}$ *is a total labelling function, and* $w_0 \in W$ *is the initial state.*

For all $w \in W$, $\mu(w)$ is the set of proposition letters which hold at that state, while $\delta$ is the transition relation which constrains the evolution of the system over time.

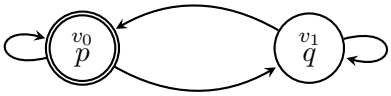



Figure 2: The Kripke structure $\mathcal{K}_2$.

Figure 2 depicts a Kripke structure, $\mathcal{K}_2$, with two states (the initial state is identified by a double circle). Formally, $\mathcal{K}_2$ is defined by the following quintuple:

$$(\{p, q\}, \{v_0, v_1\}, \{(v_0, v_0), (v_0, v_1), (v_1, v_0), (v_1, v_1)\}, \mu, v_0),$$

where $\mu(v_0) = \{p\}$ and $\mu(v_1) = \{q\}$.

**Definition 3.** *A track* $\rho$ *over a finite Kripke structure* $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ *is a* finite *sequence of states* $v_0 \cdots v_n$*, with* $n \geq 1$*, such that for all* $i \in \{0, \cdots, n-1\}$*,* $(v_i, v_{i+1}) \in \delta$*.*

Let $\mathrm{Trk}_{\mathcal{K}}$ be the (infinite) set of all tracks over a finite Kripke structure $\mathcal{K}$. For any track $\rho = v_0 \cdots v_n \in \mathrm{Trk}_{\mathcal{K}}$, we define:

- $|\rho| = n + 1$;
- $\rho(i) = v_i$, for $0 \leq i \leq |\rho| - 1$;
- $\mathrm{states}(\rho) = \{v_0, \cdots, v_n\} \subseteq W$;
- $\mathrm{intstates}(\rho) = \{v_1, \cdots, v_{n-1}\} \subseteq W$;
- $\mathrm{fst}(\rho) = v_0$ and $\mathrm{lst}(\rho) = v_n$;
- $\rho(i, j) = v_i \cdots v_j$ is a subtrack of $\rho$, for $0 \leq i < j \leq |\rho| - 1$;
- $\mathrm{Pref}(\rho) = \{\rho(0, i) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper prefixes of $\rho$. Note that $\mathrm{Pref}(\rho) = \emptyset$ if $|\rho| = 2$;
- $\mathrm{Suff}(\rho) = \{\rho(i, |\rho| - 1) \mid 1 \leq i \leq |\rho| - 2\}$ is the set of all proper suffixes of $\rho$. Note that $\mathrm{Suff}(\rho) = \emptyset$ if $|\rho| = 2$.

It is worth pointing out that the length of tracks, prefixes, and suffixes is greater than 1, as they will be mapped into strict intervals. If $\mathrm{fst}(\rho) = w_0$ (the initial state of $\mathcal{K}$), $\rho$ is said to be an *initial track*. In the following, we will denote by $\rho \cdot \rho'$ the concatenation of the tracks $\rho$ and $\rho'$, assuming that $(\mathrm{lst}(\rho), \mathrm{fst}(\rho')) \in \delta$ hence $\rho \cdot \rho' \in \mathrm{Trk}_{\mathcal{K}}$; moreover, by $\rho^n$ we will denote the track obtained by concatenating $n$ copies of $\rho$.

An abstract interval model (over $\mathrm{Trk}_{\mathcal{K}}$) can be naturally associated with a finite Kripke structure by interpreting every track as an interval bounded by its first and last states.

**Definition 4** ([20])**.** *The abstract interval model induced by a finite Kripke structure* $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ *is the abstract interval model* $\mathcal{A}_{\mathcal{K}} = (\mathcal{AP}, \mathbb{I}, A_{\mathbb{I}}, B_{\mathbb{I}}, E_{\mathbb{I}}, \sigma)$*, where:*

- $\mathbb{I} = \mathrm{Trk}_{\mathcal{K}}$,
- $A_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \mathrm{lst}(\rho) = \mathrm{fst}(\rho')\}$,
- $B_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Pref}(\rho)\}$,
- $E_{\mathbb{I}} = \{(\rho, \rho') \in \mathbb{I} \times \mathbb{I} \mid \rho' \in \mathrm{Suff}(\rho)\}$*, and*
- $\sigma : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ *where* $\sigma(\rho) = \bigcap_{w \in \mathrm{states}(\rho)} \mu(w)$*, for all* $\rho \in \mathbb{I}$.

In Definition 4, relations $A_{\mathbb{I}}, B_{\mathbb{I}}$, and $E_{\mathbb{I}}$ are interpreted as Allen's interval relations *meets*, *started-by*, and *finished-by*, respectively. Moreover, according to the definition of $\sigma$, a proposition letter $p \in \mathcal{AP}$ holds over $\rho = v_0 \cdots v_n$ if and only if it holds over all the states $v_0, \ldots, v_n$ of $\rho$. This conforms to the *homogeneity principle*, according to which a proposition letter holds over an interval if and only if it holds over all of its subintervals.

*Satisfiability* of an HS formula over a finite Kripke structure can be given in terms of induced abstract interval models.

**Definition 5.** *Let* $\mathcal{K}$ *be a finite Kripke structure,* $\rho$ *be a track in* $\mathrm{Trk}_{\mathcal{K}}$*, and* $\psi$ *be an HS formula. We say that the pair* $(\mathcal{K}, \rho)$ *satisfies* $\psi$*, denoted by* $\mathcal{K}, \rho \models \psi$*, if and only if it holds that* $\mathcal{A}_{\mathcal{K}}, \rho \models \psi$.

**Definition 6.** *Let $\mathcal{K}$ be a finite Kripke structure and $\psi$ be an HS formula. We say that $\mathcal{K}$ models $\psi$, denoted by $\mathcal{K} \models \psi$, if and only if* for all initial tracks $\rho \in \mathrm{Trk}_{\mathcal{K}}$, *it holds that $\mathcal{K}, \rho \models \psi$.*

The *model checking problem* for HS over finite Kripke structures is the problem of deciding whether $\mathcal{K} \models \psi$. Since Kripke structures feature an infinite number of tracks, the problem is not trivially decidable.

We end the section by providing some meaningful examples of properties of tracks and/or transition systems that can be expressed in HS.

**Example 1.** *The formula $[B]\bot$ can be used to select all and only the tracks of length 2. Given any $\rho$, with $|\rho| = 2$, independently of $\mathcal{K}$, it indeed holds that $\mathcal{K}, \rho \models [B]\bot$, because $\rho$ has no (strict) prefixes. On the other hand, it holds that $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle \top$ if (and only if) $|\rho| > 2$. Finally, let $\ell(k)$ be a shorthand for $[B]^{k-1}\bot \wedge \langle \mathrm{B} \rangle^{k-2} \top$. It holds that $\mathcal{K}, \rho \models \ell(k)$ if and only if $|\rho| = k$.*

**Example 2.** *Let us consider the finite Kripke structure $\mathcal{K}_2$ depicted in Figure 2. The truth of the following statements can be easily checked:*

- $\mathcal{K}_2, (v_0 v_1)^2 \models \langle \mathrm{A} \rangle\, q$;
- $\mathcal{K}_2, v_0 v_1 v_0 \not\models \langle \mathrm{A} \rangle\, q$;
- $\mathcal{K}_2, (v_0 v_1)^2 \models \langle \overline{\mathrm{A}} \rangle\, p$;
- $\mathcal{K}_2, v_1 v_0 v_1 \not\models \langle \overline{\mathrm{A}} \rangle\, p$.

*The above statements show that modalities $\langle \mathrm{A} \rangle$ and $\langle \overline{\mathrm{A}} \rangle$ can be used to distinguish between tracks that start or end at different states. In particular, note that $\langle \mathrm{A} \rangle$ (resp., $\langle \overline{\mathrm{A}} \rangle$) allows one to "move" to* any *track branching on the right (resp., left) of the considered one, e.g., if $\rho = v_0 v_1 v_0$, then $\rho\, A_{\mathbb{I}}\, v_0 v_0$, $\rho\, A_{\mathbb{I}}\, v_0 v_1$, $\rho\, A_{\mathbb{I}}\, v_0 v_0 v_0$, $\rho\, A_{\mathbb{I}}\, v_0 v_0 v_1$, $\rho\, A_{\mathbb{I}}\, v_0 v_1 v_0 v_1$, and so on.*

*Modalities $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ can be used to distinguish between tracks encompassing a different number of iterations of a given loop. This is the case, for instance, with the following statements:*

- $\mathcal{K}_2, (v_1 v_0)^3 v_1 \models \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle\, (\langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle\, \langle \mathrm{A} \rangle\, p) \big)$;
- $\mathcal{K}_2, (v_1 v_0)^2 v_1 \not\models \langle \mathrm{B} \rangle \big( \langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle\, (\langle \mathrm{A} \rangle\, p \wedge \langle \mathrm{B} \rangle\, \langle \mathrm{A} \rangle\, p) \big)$.

*Finally, HS makes it possible to distinguish between $\rho_1 = v_0^3 v_1 v_0$ and $\rho_2 = v_0 v_1 v_0^3$, which feature the same number of iterations of the same loops, but differ in the order of loop occurrences: $\mathcal{K}_2, \rho_1 \models \langle \mathrm{B} \rangle\, (\langle \mathrm{A} \rangle\, q \wedge \langle \mathrm{B} \rangle\, \langle \mathrm{A} \rangle\, p)$ but $\mathcal{K}_2, \rho_2 \not\models \langle \mathrm{B} \rangle\, (\langle \mathrm{A} \rangle\, q \wedge \langle \mathrm{B} \rangle\, \langle \mathrm{A} \rangle\, p)$.*

**Example 3.** *In Figure 3, we give an example of a finite Kripke structure $\mathcal{K}_{Sched}$ that models the behaviour of a scheduler serving three processes which are continuously requesting the use of a common resource. The initial state is $v_0$: no process is served in that state. In any other state $v_i$ and $\overline{v}_i$, with $i \in \{1, 2, 3\}$, the $i$-th process is served (this is denoted by the fact that $p_i$ holds in those states). For the sake of readability, edges are marked either by $r_i$, for request$(i)$, or by $u_i$, for unlock$(i)$. However, edge labels do not have a semantic value, i.e., they are neither part of the structure definition, nor proposition letters; they are simply used to ease reference to edges. Process $i$ is served in state $v_i$, then, after "some time", a transition $u_i$ from $v_i$ to $\overline{v}_i$ is taken; subsequently, process $i$ cannot be served again immediately, as $v_i$ is not directly reachable from $\overline{v}_i$ (the scheduler cannot serve the same process twice in two successive rounds). A transition $r_j$, with $j \neq i$, from*

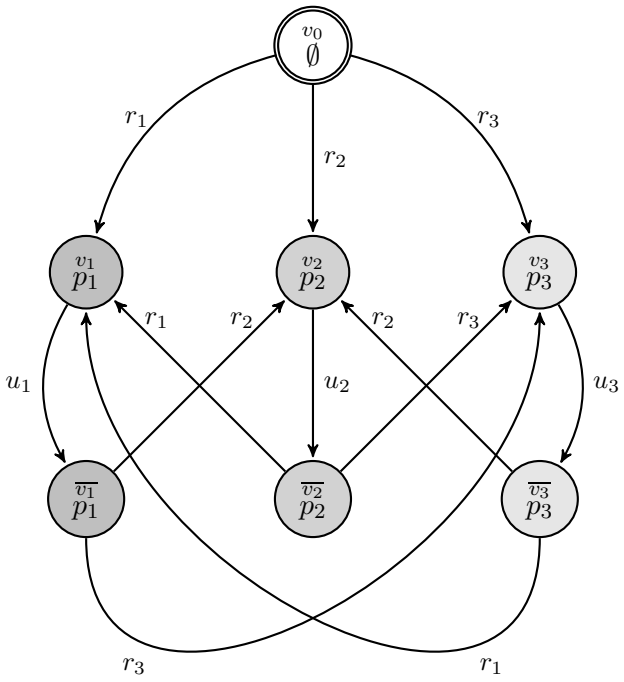



Figure 3: The Kripke structure $\mathcal{K}_{Sched}$.

$\overline{v}_i$ *to* $v_j$ *is then taken and process* $j$ *is served. This structure can be easily generalised to a higher number of processes.*

*We show how some meaningful properties to check against* $\mathcal{K}_{Sched}$ *can be expressed in HS, and, in particular, by means of formulas of the fragment* $\overline{\mathsf{A}}\mathsf{E}$*—a subfragment of the fragment* $\mathsf{A}\overline{\mathsf{A}}\mathsf{E}\overline{\mathsf{B}}\overline{\mathsf{E}}$*, on which we will focus in the following. In all formulas, we force the validity of the considered property over all legal computation sub-intervals by using modality* $[E]$ *(all computation sub-intervals are suffixes of at least one initial track). Truth of the following statements can be easily checked:*

- $\mathcal{K}_{Sched} \models [E]\big(\langle \mathrm{E}\rangle^4 \top \rightarrow (\chi(p_1,p_2) \vee \chi(p_1,p_3) \vee \chi(p_2,p_3))\big)$, *with* $\chi(p,q) := \langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle p \wedge \langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle q$*;*
- $\mathcal{K}_{Sched} \not\models [E](\langle \mathrm{E}\rangle^{10} \top \rightarrow \langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle p_3)$*;*
- $\mathcal{K}_{Sched} \not\models [E](\langle \mathrm{E}\rangle^{6} \rightarrow (\langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle p_1 \wedge \langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle p_2 \wedge \langle \mathrm{E}\rangle \langle \overline{\mathrm{A}}\rangle p_3))$.

*The first formula requires that in any suffix of length at least 6 of an initial track, at least 2 proposition letters are witnessed.* $\mathcal{K}_{Sched}$ *satisfies the formula since a process cannot be executed twice consecutively.*

*The second formula requires that in any suffix of length at least 12 of an initial track, process 3 is executed at least once in some internal states.* $\mathcal{K}_{Sched}$ *does not satisfy the formula since the scheduler, being unfair, can avoid executing a process ad libitum.*

*The third formula requires that in any suffix of length at least 8 of an initial track,* $p_1$*,* $p_2$*, and* $p_3$ *are all witnessed. The only way to satisfy this property would be to constrain the scheduler to execute the three processes in a strictly periodic manner, which is not the case.*

## 3. The notion of $B_k$-descriptor

For any finite Kripke structure $\mathcal{K}$, one can find a corresponding induced abstract interval model $\mathcal{A}_\mathcal{K}$, featuring one interval for each track of $\mathcal{K}$. As we already pointed out, since $\mathcal{K}$ has loops (each state must have at least one successor, as the transition relation $\delta$ is left-total), the number of its tracks, and thus the number of intervals of $\mathcal{A}_\mathcal{K}$, is infinite.

In [20], Molinari et al. showed that, given a bound $k$ on the structural complexity of HS formulas (that is, on the nesting depth of $\langle \mathrm{B} \rangle$ and $\langle \mathrm{E} \rangle$ modalities), it is possible to obtain a *finite* representation for $\mathcal{A}_\mathcal{K}$, which is equivalent to $\mathcal{A}_\mathcal{K}$ with respect to satisfiability of HS formulas with structural complexity less than or equal to $k$. By making use of such a representation, they prove that the model checking problem for (full) HS is decidable (with a non-elementary upper bound).

In this paper, we first restrict our attention to $\mathsf{A\overline{A}B\overline{BE}}$ and provide a model checking algorithm of lower complexity. All the results we are going to prove hold also for the fragment $\mathsf{A\overline{A}E\overline{BE}}$ by symmetry. We start with the definition of some basic notions.

**Definition 7.** *Let $\psi$ be an $\mathsf{A\overline{A}B\overline{BE}}$ formula. The B-nesting depth of $\psi$, denoted by $\mathrm{Nest}_\mathrm{B}(\psi)$, is defined by induction on the complexity of the formula as follows:*

- $\mathrm{Nest}_\mathrm{B}(p) = 0$, *for any proposition letter $p \in \mathcal{AP}$;*
- $\mathrm{Nest}_\mathrm{B}(\neg\psi) = \mathrm{Nest}_\mathrm{B}(\psi)$;
- $\mathrm{Nest}_\mathrm{B}(\psi \wedge \phi) = \max\{\mathrm{Nest}_\mathrm{B}(\psi), \mathrm{Nest}_\mathrm{B}(\phi)\}$;
- $\mathrm{Nest}_\mathrm{B}(\langle \mathrm{B} \rangle \psi) = 1 + \mathrm{Nest}_\mathrm{B}(\psi)$;
- $\mathrm{Nest}_\mathrm{B}(\langle \mathrm{X} \rangle \psi) = \mathrm{Nest}_\mathrm{B}(\psi)$, *for $X \in \{A, \overline{A}, \overline{B}, \overline{E}\}$.*

Making use of Definition 7, we can introduce the relation(s) of $k$-equivalence over tracks.

**Definition 8.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_\mathcal{K}$, and $k \in \mathbb{N}$. We say that $\rho$ and $\rho'$ are $k$-equivalent if and only if, for every $\mathsf{A\overline{A}B\overline{BE}}$ formula $\psi$ with $\mathrm{Nest}_\mathrm{B}(\psi) = k$, $\mathcal{K}, \rho \models \psi$ if and only if $\mathcal{K}, \rho' \models \psi$.*

It can be easily proved that $k$-equivalence propagates downwards.

**Proposition 9.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_\mathcal{K}$, and $k \in \mathbb{N}$. If $\rho$ and $\rho'$ are $k$-equivalent, then they are $h$-equivalent, for all $0 \leq h \leq k$.*

*Proof.* Let us assume that $\mathcal{K}, \rho \models \psi$, with $0 \leq \mathrm{Nest}_\mathrm{B}(\psi) = h \leq k$. Consider the formula $\langle \mathrm{B} \rangle^k \top$, whose B-nesting depth is equal to $k$. It holds that either $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle^k \top$ or $\mathcal{K}, \rho \models \neg \langle \mathrm{B} \rangle^k \top$. In the first case, we have that $\mathcal{K}, \rho \models \langle \mathrm{B} \rangle^k \top \wedge \psi$. Since $\mathrm{Nest}_\mathrm{B}(\langle \mathrm{B} \rangle^k \top \wedge \psi) = k$, from the hypothesis, it immediately follows that $\mathcal{K}, \rho' \models \langle \mathrm{B} \rangle^k \top \wedge \psi$, and thus $\mathcal{K}, \rho' \models \psi$. The other case can be dealt with in a symmetric way. □

We are now ready to define the key notion of *descriptor* for a track of a Kripke structure.

**Definition 10** ([20])**.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho \in \mathrm{Trk}_\mathcal{K}$, and $k \in \mathbb{N}$. The $B_k$-descriptor for $\rho$ is a labelled tree $\mathcal{D} = (V, E, \lambda)$ of depth $k$, where $V$ is a finite set of vertices, $E \subseteq V \times V$ is a set of edges, and $\lambda : V \mapsto W \times 2^W \times W$ is a node labelling function, inductively defined as follows:*

- *for $k = 0$, the $B_k$-descriptor for $\rho$ is the tree $\mathcal{D} = (\{\text{root}(\mathcal{D})\}, \emptyset, \lambda)$, where $\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$;*
- *for $k > 0$, the $B_k$-descriptor for $\rho$ is the tree $\mathcal{D} = (V, E, \lambda)$, where $\lambda(\text{root}(\mathcal{D})) = (\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$, which satisfies the following conditions:*
  1. *for each prefix $\rho'$ of $\rho$, there exists $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E$ and the subtree rooted in $v$ is the $B_{k-1}$-descriptor for $\rho'$;*
  2. *for each vertex $v \in V$ such that $(\text{root}(\mathcal{D}), v) \in E$, there exists a prefix $\rho'$ of $\rho$ such that the subtree rooted in $v$ is the $B_{k-1}$-descriptor for $\rho'$;*
  3. *for all pairs of edges $(\text{root}(\mathcal{D}), v'), (\text{root}(\mathcal{D}), v'') \in E$, if the subtree rooted in $v'$ is isomorphic to the subtree rooted in $v''$, then $v' = v''$* [1].

Condition 3 of Definition 10 simply states that no two subtrees whose roots are siblings can be isomorphic. A $B_0$-descriptor $\mathcal{D}$ for a track consists of its root only, which is denoted by $\text{root}(\mathcal{D})$. A label of a node will be referred to as a *descriptor element*: the notion of descriptor element bears analogies with an abstraction technique for discrete time Duration Calculus proposed by Hansen et al. in [14], which, on its turn, is connected to Parikh images [29] (a descriptor element can be seen as a qualitative analogue of this).

Basically, for any $k \geq 0$, the label of the root of the $B_k$-descriptor $\mathcal{D}$ for $\rho$ is the triple $(\text{fst}(\rho), \text{intstates}(\rho), \text{lst}(\rho))$. Each prefix $\rho'$ of $\rho$ is associated with some subtree whose root is labelled with $(\text{fst}(\rho'), \text{intstates}(\rho'), \text{lst}(\rho'))$ and is a child of the root of $\mathcal{D}$. Such a construction is then iteratively applied to the children of the root until either depth $k$ is reached or a track of length 2 is being considered on a node.

Hereafter equality between descriptors is considered *up to isomorphism*.

As an example, in Figure 4 we show the $B_2$-descriptor for the track $\rho = v_0v_1v_0v_0v_0v_0v_1$ of $\mathcal{K}_2$ (Figure 2). It is worth noting that there exist two distinct prefixes of $\rho$, that is, the tracks $\rho' = v_0v_1v_0v_0v_0v_0$ and $\rho'' = v_0v_1v_0v_0v_0$, which have the same $B_1$-descriptor. Since, according to Definition 10, no tree can occur more than once as a subtree of the same node (in this example, the root), in the $B_2$-descriptor for $\rho$, prefixes $\rho'$ and $\rho''$ are represented by the same tree (the first subtree of the root on the left). This shows that, in general, the root of a descriptor for a track with $h$ proper prefixes does not necessarily have $h$ children.

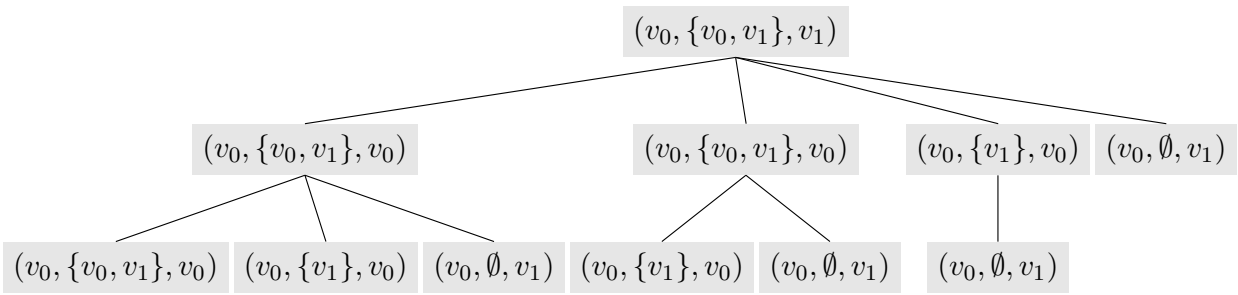



Figure 4: The $B_2$-descriptor for the track $v_0v_1v_0v_0v_0v_0v_1$ of $\mathcal{K}_2$.

[1] Here and in the following, we write subtree for maximal subtree. Moreover, isomorphism between descriptors accounts for node labels, as well (not only for the structure of descriptors).

$B$-descriptors do not convey, in general, enough information to determine which track they were built from; however, they can be used to determine which $\mathsf{A\overline{A}B\overline{B}E}$ formulas are satisfied by the track from which they were built.

In [20], the authors prove that, for a finite Kripke structure $\mathcal{K}$, there exists a *finite number* (non-elementary w.r.t. $|W|$ and $k$) of possible $B_k$-descriptors. Moreover, the number of nodes of a descriptor has a non-elementary upper bound as well. Since the number of tracks of $\mathcal{K}$ is infinite, and for any $k \in \mathbb{N}$ the set of $B_k$-descriptors for its tracks is finite, at least one $B_k$-descriptor must be the $B_k$-descriptor of *infinitely many* tracks. Thus, $B_k$-descriptors naturally induce an equivalence relation of finite index over the set of tracks of a finite Kripke structure (*$k$-descriptor equivalence relation*).

**Definition 11.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$, and $k \in \mathbb{N}$. We say that $\rho$ and $\rho'$ are* $k$-descriptor equivalent *(denoted as $\rho \sim_k \rho'$) if and only if the $B_k$-descriptors for $\rho$ and $\rho'$ coincide.*

**Lemma 12.** *Let $k \in \mathbb{N}$, $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure and $\rho_1$, $\rho_1'$, $\rho_2$, $\rho_2'$ be tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $(\mathrm{lst}(\rho_1), \mathrm{fst}(\rho_1')) \in \delta$, $(\mathrm{lst}(\rho_2), \mathrm{fst}(\rho_2')) \in \delta$, $\rho_1 \sim_k \rho_2$ and $\rho_1' \sim_k \rho_2'$. Then $\rho_1 \cdot \rho_1' \sim_k \rho_2 \cdot \rho_2'$.*

The proof is reported in Appendix A.1. The next proposition immediately follows from Lemma 12.

**Proposition 13** (Left and right extensions)**.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, v_0)$ be a finite Kripke structure, $\rho, \rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$ such that $\rho \sim_k \rho'$, and $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$. If $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$, then $\rho \cdot \overline{\rho} \sim_k \rho' \cdot \overline{\rho}$, and if $(\mathrm{lst}(\overline{\rho}), \mathrm{fst}(\rho)) \in \delta$, then $\overline{\rho} \cdot \rho \sim_k \overline{\rho} \cdot \rho'$.*

The next theorem proves that, for any pair of tracks $\rho, \rho' \in \mathrm{Trk}_{\mathcal{K}}$, if $\rho \sim_k \rho'$, then $\rho$ and $\rho'$ are $k$-equivalent (see Definition 8).

**Theorem 14** ([20])**.** *Let $\mathcal{K}$ be a finite Kripke structure, $\rho$ and $\rho'$ be two tracks in $\mathrm{Trk}_{\mathcal{K}}$, and $\psi$ be a formula of $\mathsf{A\overline{A}B\overline{B}E}$ with $\mathrm{Nest}_{\mathrm{B}}(\psi) = k$. If $\rho \sim_k \rho'$, then $\mathcal{K}, \rho \models \psi \iff \mathcal{K}, \rho' \models \psi$.*

Since the set of $B_k$-descriptors for the tracks of a finite Kripke structure $\mathcal{K}$ is finite, i.e., the equivalence relation $\sim_k$ has a finite index, there always exists a finite number of $B_k$-descriptors that "satisfy" an $\mathsf{A\overline{A}B\overline{B}E}$ formula $\psi$ with $\mathrm{Nest}_{\mathrm{B}}(\psi) = k$ (this can be formally proved by a quotient construction [20]).

## 4. Clusters and descriptor element indistinguishability

A $B_k$-descriptor provides a finite encoding for a possibly infinite set of tracks (the tracks associated with that descriptor). Unfortunately, the representation of $B_k$-descriptors as trees labelled over descriptor elements is highly redundant. For instance, given any pair of subtrees rooted in some children of the root of a descriptor, it is always the case that one of them is a subtree of the other: the two subtrees are associated with two (different) prefixes of a track and one of them is necessarily a prefix of the other. In practice, the size of the tree representation of $B_k$-descriptors prevents their direct use in model checking algorithms, and makes it difficult to determine the intrinsic complexity of $B_k$-descriptors.

In this section, we devise a more compact representation of $B_k$-descriptors. Each class of the $k$-descriptor equivalence relation is a set of $k$-equivalent tracks. For any such class, we select (at

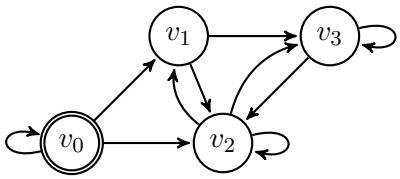



Figure 5: An example of finite Kripke structure.

least) one track representative whose length is (exponentially) bounded in both the size of $W$ (the set of states of the Kripke structure) and $k$. In order to determine such a bound, we consider suitable ordered sequences (possibly with repetitions) of descriptor elements of a $B_k$-descriptor. Let the *descriptor sequence* for a track be the ordered sequence of descriptor elements associated with its prefixes. It can be easily checked that in a descriptor sequence descriptor elements can be repeated. We devise a criterion to avoid such repetitions whenever they cannot be distinguished by an $\mathsf{A\overline{A}B\overline{B}\overline{E}}$ formula of $B$-nesting depth up to $k$.

**Definition 15.** *Let $\rho = v_0 v_1 \cdots v_n$ be a track of a finite Kripke structure. The descriptor sequence $\rho_{ds}$ for $\rho$ is $d_0 \cdots d_{n-1}$, where $d_i = \rho_{ds}(i) = (v_0, \mathrm{intstates}(v_0 \cdots v_{i+1}), v_{i+1})$, for $i \in \{0, \ldots, n-1\}$. We denote by $DElm(\rho_{ds})$ the set of descriptor elements occurring in $\rho_{ds}$.*

As an example, let us consider the finite Kripke structure of Figure 5 and the track $\rho = v_0v_0v_0v_1v_2v_1v_2v_3v_3v_2v_2$. The descriptor sequence for $\rho$ is:

$$\rho_{ds} = (v_0, \emptyset, v_0)\boxed{(v_0, \{v_0\}, v_0)}(v_0, \{v_0\}, v_1)(v_0, \{v_0, v_1\}, v_2)\boxed{(v_0, \Gamma, v_1)(v_0, \Gamma, v_2)}$$
$$(v_0, \Gamma, v_3)\boxed{(v_0, \Delta, v_3)(v_0, \Delta, v_2)(v_0, \Delta, v_2)}, \quad (*)$$

where $\Gamma = \{v_0, v_1, v_2\}$, $\Delta = \{v_0, v_1, v_2, v_3\}$, and $DElm(\rho_{ds})$ is the set $\{(v_0, \emptyset, v_0), (v_0, \{v_0\}, v_0), (v_0, \{v_0\}, v_1), (v_0, \{v_0, v_1\}, v_2), (v_0, \Gamma, v_1), (v_0, \Gamma, v_2), (v_0, \Gamma, v_3), (v_0, \Delta, v_2), (v_0, \Delta, v_3)\}$. The meaning of boxes in (*) will be clear later.

To express the relationships between descriptor elements occurring in a descriptor sequence, we introduce a binary relation $\mathrm{R_t}$. Intuitively, given two descriptor elements $d'$ and $d''$ of a descriptor sequence, the relation $d' \,\mathrm{R_t}\, d''$ holds if $d'$ and $d''$ are the descriptor elements of two tracks $\rho'$ and $\rho''$, respectively, and $\rho'$ is a prefix of $\rho''$.

**Definition 16.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ and let $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v_{in}, S'', v''_{fin})$ be two descriptor elements in $\rho_{ds}$. It holds that $d' \,\mathrm{R_t}\, d''$ if and only if $S' \cup \{v'_{fin}\} \subseteq S''$.*

Note that the relation $\mathrm{R_t}$ is transitive. In fact for all descriptor elements $d' = (v_{in}, S', v'_{fin})$, $d'' = (v_{in}, S'', v''_{fin})$ and $d''' = (v_{in}, S''', v'''_{fin})$, if $d' \,\mathrm{R_t}\, d''$ and $d'' \,\mathrm{R_t}\, d'''$, then $S' \cup \{v'_{fin}\} \subseteq S''$ and $S'' \cup \{v''_{fin}\} \subseteq S'''$; it follows that $S' \cup \{v'_{fin}\} \subseteq S'''$, and thus $d' \,\mathrm{R_t}\, d'''$. The relation $\mathrm{R_t}$ is neither an equivalence relation nor a quasiorder, since $\mathrm{R_t}$ is neither reflexive (e.g., $(v_0, \{v_0\}, v_1) \not\mathrm{R_t} (v_0, \{v_0\}, v_1)$), nor symmetric (e.g., $(v_0, \{v_0\}, v_1)\,\mathrm{R_t}(v_0, \{v_0, v_1\}, v_1)$ and $(v_0, \{v_0, v_1\}, v_1) \not\mathrm{R_t} (v_0, \{v_0\}, v_1)$), nor antisymmetric (e.g., $(v_0, \{v_1, v_2\}, v_1)\,\mathrm{R_t}(v_0, \{v_1, v_2\}, v_2)$ and $(v_0, \{v_1, v_2\}, v_2)\,\mathrm{R_t}(v_0, \{v_1, v_2\}, v_1)$, but the two elements are distinct).

It can be easily shown that $\mathrm{R_t}$ pairs descriptor elements of increasing prefixes of a track.

**Proposition 17.** *Let $\rho_{ds}$ be the descriptor sequence for the track $\rho = v_0 v_1 \cdots v_n$. Then, $\rho_{ds}(i)\,\mathrm{R_t}\,\rho_{ds}(j)$, for all $0 \leq i < j < n$.*

We now partition descriptor elements into two different types.

**Definition 18.** *A descriptor element $(v_{in}, S, v_{fin})$ is a Type-1 descriptor element if $v_{fin} \notin S$, while it is a Type-2 descriptor element if $v_{fin} \in S$.*

It can be easily checked that a descriptor element $d = (v_{in}, S, v_{fin})$ is Type-1 if and only if $\mathrm{R_t}$ is not reflexive for $d$. In fact, if $d \not\mathrel{\mathrm{R_t}} d$, then $S \cup \{v_{fin}\} \not\subseteq S$, and thus $v_{fin} \notin S$. Conversely, if $v_{fin} \notin S$, then $d \not\mathrel{\mathrm{R_t}} d$. It follows that a Type-1 descriptor element cannot occur more than once in a descriptor sequence. On the other hand, Type-2 descriptor elements may occur multiple times, and if a descriptor element occurs more than once in a descriptor sequence, then it is necessarily of Type-2.

**Proposition 19.** *If both $d' \,\mathrm{R_t}\, d''$ and $d'' \,\mathrm{R_t}\, d'$, for $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v_{in}, S'', v''_{fin})$, then $v'_{fin} \in S'$, $v''_{fin} \in S''$, and $S' = S''$; thus, both $d'$ and $d''$ are Type-2 descriptor elements.*

We are now ready to give a general characterization of the descriptor sequence $\rho_{ds}$ for a track $\rho$: $\rho_{ds}$ is composed of some (maximal) subsequences, consisting of occurrences of Type-2 descriptor elements on which $\mathrm{R_t}$ is symmetric, separated by occurrences of Type-1 descriptor elements. This can be formalized by means of the following notion of cluster.

**Definition 20.** *A cluster $\mathcal{C}$ of (Type-2) descriptor elements is a maximal set of descriptor elements $\{d_1, \ldots, d_s\} \subseteq DElm(\rho_{ds})$ such that $d_i \,\mathrm{R_t}\, d_j$ and $d_j \,\mathrm{R_t}\, d_i$ for all $i, j \in \{1, \ldots, s\}$.*

Thanks to maximality, clusters are pairwise disjoint: if $\mathcal{C}$ and $\mathcal{C}'$ are distinct clusters, $d \in \mathcal{C}$ and $d' \in \mathcal{C}'$, either $d \,\mathrm{R_t}\, d'$ and $d' \not\mathrel{\mathrm{R_t}} d$, or $d' \,\mathrm{R_t}\, d$ and $d \not\mathrel{\mathrm{R_t}} d'$.

It can be easily checked that the descriptor elements of a cluster $\mathcal{C}$ are contiguous in $\rho_{ds}$ (in other words, they form a subsequence of $\rho_{ds}$), that is, occurrences of descriptor elements of $\mathcal{C}$ are never shuffled with occurrences of descriptor elements not belonging to $\mathcal{C}$.

**Definition 21.** *Let $\rho_{ds}$ be a descriptor sequence and $\mathcal{C}$ be one of its clusters. The subsequence of $\rho_{ds}$ associated with $\mathcal{C}$ is the subsequence $\rho_{ds}(i, j)$, with $i \leq j < |\rho_{ds}|$, including all and only the occurrences of the descriptor elements in $\mathcal{C}$.*

Note that two subsequences associated with two distinct clusters $\mathcal{C}$ and $\mathcal{C}'$ in a descriptor sequence must be separated by at least one occurrence of a Type-1 descriptor element. For instance, with reference to the descriptor sequence (*) for the track $\rho = v_0v_0v_0v_1v_2v_1v_2v_3v_3v_2v_2$ of the Kripke structure in Figure 5, the subsequences associated with clusters are enclosed in boxes.

While $\mathrm{R_t}$ allows us to order any pair of Type-1 descriptor elements, as well as any Type-1 descriptor element with respect to a Type-2 one, it does not give us any means to order Type-2 descriptor elements belonging to the same cluster. This, together with the fact that Type-2 elements may have multiple occurrences in a descriptor sequence, implies that we need to somehow limit the number of occurrences of Type-2 elements in order to determine a bound on the length of track representatives of $B_k$-descriptors.

To this end, we introduce an equivalence relation that allows us to put together indistinguishable occurrences of the same descriptor element in a descriptor sequence, that is, to detect those occurrences which are associated with prefixes of the track with the same $B_k$-descriptor. The idea is that a track representative for a $B_k$-descriptor should not feature indistinguishable occurrences of the same descriptor element.

**Definition 22.** *Let $\rho_{ds}$ be a descriptor sequence and $k \geq 1$. We say that $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, are $k$-indistinguishable if (and only if) they are occurrences* of the same descriptor element $d$ *and:*

- *(for $k = 1$) $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$;*
- *(for $k \geq 2$) for all $i \leq \ell \leq j-1$, there exists $0 \leq \ell' \leq i-1$ such that $\rho_{ds}(\ell)$ and $\rho_{ds}(\ell')$ are $(k-1)$-indistinguishable.*

From Definition 22, it follows that two indistinguishable occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$ of the same descriptor element necessarily belong to the same subsequence of $\rho_{ds}$ associated with a cluster.

In general, it is always the case that $DElm(\rho_{ds}(0, i-1)) \subseteq DElm(\rho_{ds}(0, j-1))$ for $i < j$. Moreover, note that the two first occurrences of a descriptor element, say $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are never 1-indistinguishable as a consequence of the fact that 1-indistinguishability requires that $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$.

Proposition 23 and 24 state some basic properties of the $k$-indistinguishability relation.

**Proposition 23.** *Let $k \geq 2$ and $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, be two $k$-indistinguishable occurrences of the same descriptor element in a descriptor sequence $\rho_{ds}$. Then, $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are also $(k-1)$-indistinguishable.*

*Proof.* The proof is by induction on $k \geq 2$.
*Base case* ($k = 2$). Let $\rho_{ds}(i)$ and $\rho_{ds}(j)$ be two 2-indistinguishable occurrences of a descriptor element $d$. By definition, for any $\rho_{ds}(i')$, with $i \leq i' < j$, an occurrence of the descriptor element $d' = \rho_{ds}(i')$ must exist before position $i$, and thus $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$. It immediately follows that $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are 1-indistinguishable.
*Inductive step* ($k \geq 3$). By definition, for all $i \leq \ell \leq j-1$, there exists $0 \leq \ell' \leq i-1$ such that $\rho_{ds}(\ell)$ and $\rho_{ds}(\ell')$ are $(k-1)$-indistinguishable. By the inductive hypothesis, $\rho_{ds}(\ell)$ and $\rho_{ds}(\ell')$ are $(k-2)$-indistinguishable, which implies that $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are $(k-1)$-indistinguishable. □

**Proposition 24.** *Let $k \geq 1$ and $\rho_{ds}(i)$ and $\rho_{ds}(m)$, with $0 \leq i < m < |\rho_{ds}|$, be two $k$-indistinguishable occurrences of the same descriptor element in a descriptor sequence $\rho_{ds}$. If $\rho_{ds}(j) = \rho_{ds}(m)$, for some $i < j < m$, then $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are also $k$-indistinguishable.*

*Proof.* For $k = 1$, we have $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, m-1))$; moreover, $DElm(\rho_{ds}(0, i-1)) \subseteq DElm(\rho_{ds}(0, j-1)) \subseteq DElm(\rho_{ds}(0, m-1))$. Thus $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, m-1)) = DElm(\rho_{ds}(0, j-1))$, proving the property.

If $k \geq 2$, all occurrences $\rho_{ds}(i')$, with $i \leq i' < m$, are $(k-1)$-indistinguishable from some occurrence of the same descriptor element before $i$, by hypothesis. In particular, this is true for all occurrences $\rho_{ds}(j')$, with $j \leq j' < m$. The thesis trivially follows. □

**Example 4.** *In Figure 6, we give some examples of $k$-indistinguishability relations, for $k \in \{1, 2, 3\}$, considering the track $\rho = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2v_2v_3v_2$ of the finite Kripke structure depicted in Figure 5. The track $\rho$ generates the descriptor sequence $\rho_{ds} = (v_0, \emptyset, v_1)(v_0, \{v_1\}, v_2)(v_0, \{v_1, v_2\}, v_3)abaababaababcababcbcabbab$, where $a$, $b$, and $c$ stand for $(v_0, \{v_1, v_2, v_3\}, v_3)$, $(v_0, \{v_1, v_2, v_3\}, v_2)$, and $(v_0, \{v_1, v_2, v_3\}, v_1)$, respectively. The figure shows the subsequence $\rho_{ds}(3, |\rho_{ds}| - 1)$ associated with the cluster $\mathcal{C} = \{a, b, c\}$. Pairs of $k$-indistinguishable consecutive occurrences of descriptor elements are connected by a rounded edge labelled by $k$. Edges labelled by $\times$ link occurrences which are not 1-indistinguishable. The values of all missing edges*

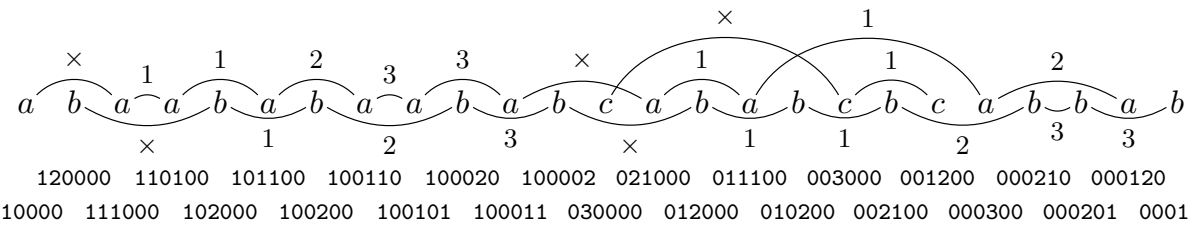



Figure 6: Examples of $k$-indistinguishability relations.

*can easily be derived using the property stated by Corollary 26 below. The meaning of numerical strings at the bottom of the figure will be clear later.*

The next theorem establishes a fundamental connection between $k$-indistinguishability of descriptor elements and $k$-descriptor equivalence of tracks.

**Theorem 25.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$. Two occurrences $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $0 \leq i < j < |\rho_{ds}|$, of the same descriptor element are $k$-indistinguishable if and only if $\rho(0, i+1) \sim_k \rho(0, j+1)$.*

*Proof.* Let us assume that $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are $k$-indistinguishable. We prove by induction on $k \geq 1$ that $\rho(0, i+1)$ and $\rho(0, j+1)$ are associated with the same $B_k$-descriptor.

*Base case* ($k = 1$). Since $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are occurrences of the same descriptor element, the roots of the $B_1$-descriptors for $\rho(0, i+1)$ and for $\rho(0, j+1)$ are labelled by the same descriptor element. Moreover, for each leaf of the $B_1$-descriptor for $\rho(0, i+1)$ there is a leaf of the $B_1$-descriptor for $\rho(0, j+1)$ with the same label, and vice versa, as by 1-indistinguishability $DElm(\rho_{ds}(0, i-1)) = DElm(\rho_{ds}(0, j-1))$.

*Inductive step* ($k \geq 2$). Since all the prefixes of $\rho(0, i+1)$ are also prefixes of $\rho(0, j+1)$, we just need to focus on the prefixes $\rho(0, t)$, with $i+1 \leq t \leq j$, in order to show that $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same $B_k$-descriptor. By definition, any occurrence $\rho_{ds}(i')$ with $i \leq i' < j$, is $(k-1)$-indistinguishable from another occurrence $\rho_{ds}(i'')$, with $i'' < i$, of the same descriptor element. By the inductive hypothesis, $\rho(0, i'+1)$ and $\rho(0, i''+1)$ are associated with the same $B_{k-1}$-descriptor. It follows that, for any proper prefix of $\rho(0, j+1)$ (of length at least 2), there exists a proper prefix of $\rho(0, i+1)$ with the same $B_{k-1}$-descriptor, which implies that the tracks $\rho(0, i+1)$ and $\rho(0, j+1)$ are associated with the same $B_k$-descriptor.

Conversely, we prove by induction on $k \geq 1$ that if $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* $k$-indistinguishable, then the $B_k$-descriptors for $\rho(0, i+1)$ and $\rho(0, j+1)$ are different. We assume $\rho_{ds}(i)$ and $\rho_{ds}(j)$ to be occurrences of the same descriptor element (if this was not the case, the thesis would trivially follow, since the roots of the $B_k$-descriptors for $\rho(0, i+1)$ and $\rho(0, j+1)$ would be labelled by different descriptor elements).

*Base case* ($k = 1$). If $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* 1-indistinguishable, $DElm(\rho_{ds}(0, i-1)) \subset DElm(\rho_{ds}(0, j-1))$. Hence, there is $d \in DElm(\rho_{ds}(0, j-1))$ such that $d \notin DElm(\rho_{ds}(0, i-1))$, and thus the $B_1$-descriptor for $\rho(0, j+1)$ has a leaf labelled by $d$ which is not present in the $B_1$-descriptor for $\rho(0, i+1)$.

*Inductive step* ($k \geq 2$). If $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, are *not* $k$-indistinguishable, then there exists (at least) one occurrence $\rho_{ds}(i')$, with $i \leq i' < j$, of a descriptor element $d$ which is *not* $(k-1)$-indistinguishable from any occurrence of $d$ before position $i$. By the inductive hypothesis,

$\rho(0, i'+1)$ is associated to a $B_{k-1}$-descriptor which is not equal to any $B_{k-1}$-descriptors associated with proper prefixes of $\rho(0, i+1)$. Thus, in the $B_k$-descriptor for $\rho(0, j+1)$ there exists a subtree of depth $k-1$ such that there is no isomorphic subtree of depth $k-1$ in the $B_k$-descriptor for $\rho(0, i+1)$. □

Note that $k$-indistinguishability between occurrences of descriptor elements is defined *only for pairs of prefixes of the same track*, while the relation of $k$-descriptor equivalence can be applied to pairs of any tracks of a Kripke structure.

The next corollary easily follows from Theorem 25.

**Corollary 26.** *Let $\rho_{ds}(i)$, $\rho_{ds}(j)$, and $\rho_{ds}(m)$, with $0 \leq i < j < m < |\rho_{ds}|$, be three occurrences of the same descriptor element in a descriptor sequence $\rho_{ds}$. If both the pair $\rho_{ds}(i)$ and $\rho_{ds}(j)$ and the pair $\rho_{ds}(j)$ and $\rho_{ds}(m)$ are $k$-indistinguishable, for some $k \geq 1$, then $\rho_{ds}(i)$ and $\rho_{ds}(m)$ are also $k$-indistinguishable.*

## 5. A model checking procedure for $\mathbf{A\overline{A}B\overline{B}E}$ based on track representatives

In this section, we will exploit the $k$-indistinguishability relation(s) between descriptor elements in a descriptor sequence $\rho_{ds}$ for a track $\rho$ to possibly replace $\rho$ by a $k$-descriptor equivalent, *shorter* track $\rho'$ of bounded length. This allows us to find, for each $B_k$-descriptor $\mathcal{D}_{B_k}$ (witnessed by a track of a finite Kripke structure $\mathcal{K}$), a *track representative* $\tilde{\rho}$ in $\mathcal{K}$ such that $(i)$ $\mathcal{D}_{B_k}$ is the $B_k$-descriptor for $\tilde{\rho}$ and $(ii)$ the length of $\tilde{\rho}$ is bounded. Thanks to property $(ii)$, we can check all the track representatives of a finite Kripke structure by simply visiting its unravelling up to a bounded depth.

The notion of track representative can be explained as follows. Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$. If there are two occurrences of the same descriptor element $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $i < j$, which are $k$-indistinguishable—let $\rho = \rho(0, j+1) \cdot \overline{\rho}$, with $\overline{\rho} = \rho(j+2, |\rho|-1)$—then we can replace $\rho$ by the $k$-descriptor equivalent, shorter track $\rho(0, i+1) \cdot \overline{\rho}$. By Theorem 25, $\rho(0, i+1)$ and $\rho(0, j+1)$ have the same $B_k$-descriptor and thus, by Proposition 13, $\rho = \rho(0, j+1) \cdot \overline{\rho}$ and $\rho(0, i+1) \cdot \overline{\rho}$ have the same $B_k$-descriptor. Moreover, since $\rho_{ds}(i)$ and $\rho_{ds}(j)$ are occurrences of the same descriptor element, $\rho(i+1) = \rho(j+1)$ and thus the track $\rho(0, i+1) \cdot \overline{\rho}$ is witnessed in the finite Kripke structure. By iteratively applying such a *contraction method*, we can find a track $\rho'$ which is $k$-descriptor equivalent to $\rho$, whose descriptor sequence is devoid of $k$-indistinguishable occurrences of descriptor elements. *A track representative* is a track that fulfils this property.

We now show how to calculate a bound to the length of track representatives. We start by stating some technical properties. The next proposition provides a bound to the distance within which we necessarily observe a repeated occurrence of some descriptor element in the descriptor sequence for a track. We preliminarily observe that, for any track $\rho$, $|DElm(\rho_{ds})| \leq 1 + |W|^2$, where $W$ is the set of states of the finite Kripke structure. Indeed, in the descriptor sequence, the sets of internal states of prefixes of $\rho$ increase monotonically with respect to the "$\subseteq$" relation. As a consequence, at most $|W|$ distinct sets may occur—excluding $\emptyset$ which can occur only in the first descriptor element. Moreover, these sets can be paired with all possible final states, which are at most $|W|$.

**Proposition 27.** *For each track $\rho$ of $\mathcal{K}$, associated with a descriptor element $d$, there exists a track $\rho'$ of $\mathcal{K}$, associated with the same descriptor element $d$, such that $|\rho'| \leq 2 + |W|^2$.*

*Proof.* By induction on the length $\ell \geq 2$ of $\rho$.
*Base case* ($\ell = 2$). The track $\rho$ satisfies the condition $\ell \leq 2 + |W|^2$.
*Inductive step* ($\ell > 2$). We distinguish two cases. If $\rho_{ds}$ has no duplicated occurrences of the same descriptor element, then $|\rho_{ds}| \leq 1 + |W|^2$, since $|DElm(\rho_{ds})| \leq 1 + |W|^2$, and thus $\ell \leq 2 + |W|^2$ (the length of $\rho$ is equal to the length of $\rho_{ds}$ plus 1).

On the other hand, if $\rho_{ds}(i) = \rho_{ds}(j)$, for some $0 \leq i < j < |\rho_{ds}|$, $\rho(0, i+1)$ and $\rho(0, j+1)$ are associated with the same descriptor element. Now, $\rho' = \rho(0, i+1) \cdot \rho(j+2, |\rho| - 1)$ is a track of $\mathcal{K}$ since $\rho(i+1) = \rho(j+1)$, and, by Proposition 13, $\rho = \rho(0, j+1) \cdot \rho(j+2, |\rho| - 1)$ and $\rho'$ are associated with the same descriptor element. By the inductive hypothesis, there exists a track $\rho''$ of $\mathcal{K}$, associated with the same descriptor element of $\rho'$ (and of $\rho$), with $|\rho''| \leq 2 + |W|^2$. □

Proposition 27 will be used in the following unravelling Algorithm 1 as a termination criterion (referred to as *0-termination criterion*) for unravelling a finite Kripke structure when it is not necessary to observe multiple occurrences of the same descriptor element: *to get a track representative for every descriptor element with initial state $v$, witnessed in a finite Kripke structure with set of states $W$, we can avoid considering tracks longer than $2 + |W|^2$ while exploring the unravelling of the Kripke structure from $v$.*

Let us now consider the (more difficult) problem of establishing a bound for tracks devoid of pairs of $k$-indistinguishable occurrences of descriptor elements. We first note that, in a descriptor sequence $\rho_{ds}$ for a track $\rho$, there are at most $|W|$ occurrences of Type-1 descriptor elements. On the other hand, Type-2 descriptor elements can occur multiple times and thus, to bound the length of $\rho_{ds}$, one has to constrain the *number* and the *length* of the subsequences of $\rho_{ds}$ associated with clusters. As for their number, it suffices to observe that they are separated by Type-1 descriptor elements, and hence at most $|W|$ of them, related to distinct clusters, can occur in a descriptor sequence.

As for their length, we can proceed as follows. First, for any cluster $\mathcal{C}$, it holds that $|\mathcal{C}| \leq |W|$, as all (Type-2) descriptor elements of $\mathcal{C}$ share the same set $S$ of internal states and their final states $v_{fin}$ must belong to $S$. In the following, we consider the (maximal) subsequence $\rho_{ds}(u, v)$ of $\rho_{ds}$ associated with a specific cluster $\mathcal{C}$, for some $0 \leq u \leq v \leq |\rho_{ds}| - 1$, and when we mention an index $i$, we implicitly assume that $u \leq i \leq v$, that is, $i$ refers to a position in the subsequence.

We sequentially scan such a subsequence suitably recording the multiplicity of occurrences of descriptor elements into an auxiliary structure. To detect indistinguishable occurrences of descriptor elements up to indistinguishability $s \geq 1$, we use $s + 3$ arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$, $Q_1()$, …, $Q_s()$. Array elements are sets of descriptor elements of $\mathcal{C}$: given an index $i$, the sets at position $i$, $Q_{-2}(i)$, $Q_{-1}(i)$, $Q_0(i)$, $Q_1(i)$, …, $Q_s(i)$, store information about indistinguishability for multiple occurrences of descriptor elements in the subsequence up to position $i > u$. To exemplify, if we find an occurrence of the descriptor element $d \in \mathcal{C}$ at position $i$, that is, $\rho_{ds}(i) = d$, we have that:

1. $Q_{-2}(i)$ contains all descriptor elements of $\mathcal{C}$ which have never occurred in $\rho_{ds}(u, i)$;
2. $d \in Q_{-1}(i)$ if $d$ has never occurred in $\rho_{ds}(u, i-1)$ and $\rho_{ds}(i) = d$, that is, $\rho_{ds}(i)$ is the first occurrence of $d$ in $\rho_{ds}(u, i)$;
3. $d \in Q_0(i)$ if $d$ occurs at least twice in $\rho_{ds}(u, i)$ and the occurrence $\rho_{ds}(i)$ of $d$ is *not* 1-indistinguishable from the last occurrence of $d$ in $\rho_{ds}(u, i-1)$;
4. $d \in Q_t(i)$ (for some $t \geq 1$) if the occurrence $\rho_{ds}(i)$ of $d$ is $t$-indistinguishable, but *not also* $(t+1)$-indistinguishable, from the last occurrence of $d$ in $\rho_{ds}(u, i-1)$.

$f(\rho_{ds}, u) = \big(\mathcal{C} \setminus \{d\}, \{d\}, \emptyset, \cdots, \emptyset\big)$ with $\rho_{ds}(u) = d$;

For all $i > u$: $f(\rho_{ds}, i) = \big(Q_{-2}(i), Q_{-1}(i), Q_0(i), \ldots, Q_s(i)\big) =$

$\begin{cases} \big(Q_{-2}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=-1}^{s} Q_m(i-1), \emptyset, \ldots, \emptyset\big) \text{ if } \rho_{ds}(i) \text{ is the first occurrence of } d \text{ in } \rho_{ds}(u,i); \textbf{(a)} \\ \big(Q_{-2}(i-1), Q_{-1}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=0}^{s} Q_m(i-1), \emptyset, \ldots, \emptyset\big) \text{ if } \rho_{ds}(i) = d,\ d \in Q_{-1}(i-1), \text{ and } \rho_{ds}(i) \text{ is at least the second occurrence of } d \text{ in } \rho_{ds}(u,i) \text{ and it is } \textit{not} \text{ 1-indistinguishable from the immediately preceding occurrence of } d; \textbf{(b)} \\ \big(Q_{-2}(i-1), Q_{-1}(i-1), \{d\} \cup Q_0(i-1), Q_1(i-1) \setminus \{d\}, \ldots, Q_s(i-1) \setminus \{d\}\big) \text{ if } \rho_{ds}(i) = d,\ d \in \bigcup_{m=0}^{s} Q_m(i-1), \text{ and } \rho_{ds}(i) \text{ is at least the second occurrence of } d \text{ in } \rho_{ds}(u,i) \text{ and it is } \textit{not} \text{ 1-indistinguishable from the immediately preceding occurrence of } d; \textbf{(c)} \\ \big(Q_{-2}(i-1) \setminus \{d\}, \ldots, Q_{t-1}(i-1) \setminus \{d\}, \{d\} \cup \bigcup_{m=t}^{s} Q_m(i-1), \emptyset, \ldots, \emptyset\big) \text{ if } \rho_{ds}(i) = d,\ \rho_{ds}(i) \text{ is } t\text{-indistinguishable (for some } t \geq 1), \text{ but } \textit{not} \text{ also } (t+1)\text{-indistinguishable, to the immediately preceding occurrence of } d, \text{ and } d \in \bigcup_{m=-2}^{t-1} Q_m(i-1); \textbf{(d)} \\ \big(Q_{-2}(i-1), \cdots, Q_{t-1}(i-1), \{d\} \cup Q_t(i-1), Q_{t+1}(i-1) \setminus \{d\}, \ldots, Q_s(i-1) \setminus \{d\}\big) \text{ if } \rho_{ds}(i) = d,\ \rho_{ds}(i) \text{ is } t\text{-indistinguishable (for some } t \geq 1), \text{ but } \textit{not} \text{ also } (t+1)\text{-indistinguishable, to the immediately preceding occurrence of } d, \text{ and } d \in \bigcup_{m=t}^{s} Q_m(i-1). \textbf{(e)} \end{cases}$

Figure 7: Definition of the scan function $f$.

In particular, at position $u$ (the first of the subsequence), $Q_{-1}(u)$ contains only the descriptor element $d = \rho_{ds}(u)$, $Q_{-2}(u)$ is the set $\mathcal{C} \setminus \{d\}$, and $Q_0(u)$, $Q_1(u)$, ... are empty sets.

Arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$, $Q_1()$, ..., $Q_s()$ satisfy the following constraints: for all positions $i$, $\bigcup_{m=-2}^{s} Q_m(i) = \mathcal{C}$ and, for all $i$ and all $m \neq m'$, $Q_m(i) \cap Q_{m'}(i) = \emptyset$.

Intuitively, at every position $i$, $Q_{-2}(i)$, $Q_{-1}(i), Q_0(i), Q_1(i)$, ..., $Q_s(i)$ describe a *state* of the scanning process of the subsequence. The change of state produced by the transition from position $i-1$ to position $i$ while scanning the subsequence is formally defined by the function $f$, reported in Figure 7, which maps the descriptor sequence $\rho_{ds}$ and a position $i$ to the tuple of sets $\big(Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \ldots, Q_s(i)\big)$.

Note that, whenever a descriptor element $\rho_{ds}(i) = d$ is such that $d \in Q_z(i-1)$ and $d \in Q_{z'}(i)$, with $z < z'$ (cases (a), (b), and (d) of the definition of $f$), all $Q_{z''}(i)$, with $z'' > z'$, are empty sets and, for all $z'' \geq z'$, all elements in $Q_{z''}(i-1)$ belong to $Q_{z'}(i)$. As an intuitive explanation, consider, for instance, the following scenario: in a subsequence of $\rho_{ds}$, associated with some cluster $\mathcal{C}$, $\rho_{ds}(h) = \rho_{ds}(i) = d \in \mathcal{C}$ and $\rho_{ds}(h') = \rho_{ds}(i') = d' \in \mathcal{C}$, for some $h < h' < i < i'$ and $d \neq d'$, and there are not other occurrences of $d$ and $d'$ in $\rho_{ds}(h, i')$. If $\rho_{ds}(h)$ and $\rho_{ds}(i)$ are exactly $z'$-indistinguishable, by definition of the indistinguishability relation, $\rho_{ds}(h')$ and $\rho_{ds}(i')$ can be no more than $(z'+1)$-indistinguishable. Thus, if $d'$ is in $Q_{z''}(i-1)$, for some $z'' > z'$, we can safely "downgrade" it to $Q_{z'}(i)$, because we know that, when we meet the next occurrence of $d'$ ($\rho_{ds}(i')$), $\rho_{ds}(h')$ and $\rho_{ds}(i')$ will be no more than $(z'+1)$-indistinguishable.

In the following, we will make use of an abstract characterization of the state of arrays at a given position $i$, as determined by the scan function $f$, called *configuration*, that only accounts for the cardinality of sets in arrays. Theorem 29 states that, when a descriptor subsequence is scanned,

configurations never repeat, since the sequence of configurations is strictly decreasing according to the lexicographical order $>_{lex}$. This property will allow us to establish the desired bound on the length of track representatives.

**Definition 28.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ and $i$ be a position in the subsequence of $\rho_{ds}$ associated with a given cluster. The* configuration at position $i$, *denoted as $c(i)$, is the tuple*

$$c(i) = (|Q_{-2}(i)|, |Q_{-1}(i)|, |Q_0(i)|, |Q_1(i)|, \cdots, |Q_s(i)|),$$

*where $f(\rho_{ds}, i) = (Q_{-2}(i), Q_{-1}(i), Q_0(i), Q_1(i), \cdots, Q_s(i))$.*

An example of a sequence of configurations is given in Figure 6 of Example 4, where, for each position in the subsequence $\rho_{ds}(3, |\rho_{ds}| - 1)$, we give the associated configuration: $c(3) = (2, 1, 0, 0, 0, 0)$, $c(4) = (1, 2, 0, 0, 0, 0)$, and so forth.

**Theorem 29.** *Let $\rho_{ds}$ be the descriptor sequence for a track $\rho$ and $\rho_{ds}(u, v)$, for some $u < v$, be the subsequence associated with a cluster $\mathcal{C}$. For all $u < i \leq v$, if $\rho_{ds}(i) = d$, then it holds that $d \in Q_t(i-1)$, $d \in Q_{t+1}(i)$, for some $t \in \{-2, -1\} \cup \mathbb{N}$, and $c(i-1) >_{lex} c(i)$.*

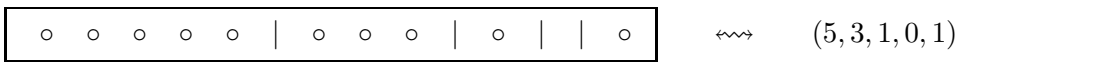
The proof is given in Appendix A.2.

We show now how to select all and only those tracks which do not feature any pair of $k$-indistinguishable occurrences of descriptor elements. To this end, we make use of a scan function $f$ which uses $k + 3$ arrays (the value $k + 3$ accounts for the parameter $k$ of descriptor element indistinguishability, plus the three arrays $Q_{-2}()$, $Q_{-1}()$, $Q_0()$). Theorem 29 guarantees that, while scanning a subsequence, configurations never repeat. This allows us to set an upper bound to the length of a track such that, whenever exceeded, the descriptor sequence for the track features at least a pair of $k$-indistinguishable occurrences of some descriptor element. The bound is essentially given by the number of possible configurations for $k + 3$ arrays.

By an easy combinatorial argument, we can prove the following proposition.

**Proposition 30.** *For all $n, t \in \mathbb{N} \setminus \{0\}$, the number of distinct $t$-tuples of natural numbers whose sum equals $n$ is $\varepsilon(n, t) = \binom{n+t-1}{n} = \binom{n+t-1}{t-1}$.*

*Proof.* The following figure suggests an alternative representation of a tuple, in the form of a configuration of separators/bullets:




It can be easily checked that such a representation is *unambiguous*, i.e., there exists a bijection between configurations of separators/bullets and tuples.

The sum of the natural numbers of the tuple equals the number of bullets, and the size of the tuple is the number of separators plus 1. Since there are $\varepsilon(n, t) = \binom{n+t-1}{t-1}$ distinct ways of choosing $t - 1$ separators among $n + t - 1$ different places—and places which are not chosen must contain bullets—there are exactly $\varepsilon(n, t)$ distinct $t$-tuples of natural numbers whose sum equals $n$. □

Proposition 30 provides two upper bounds for $\varepsilon(n, t)$: $\varepsilon(n, t) \leq (n + 1)^{t-1}$ and $\varepsilon(n, t) \leq t^n$.

Since a configuration $c(i)$ of a cluster $\mathcal{C}$ is a $(k + 3)$-tuple whose elements add up to $|\mathcal{C}|$, by Proposition 30 we conclude that there are at most $\varepsilon(|\mathcal{C}|, k + 3) = \binom{|\mathcal{C}|+k+2}{k+2}$ distinct configurations of size $(k + 3)$, whose natural numbers add up to $|\mathcal{C}|$. Moreover, since configurations never repeat

while scanning a subsequence associated with a cluster $\mathcal{C}$, $\varepsilon(|\mathcal{C}|, k+3)$ is an upper bound to the length of such a subsequence.

Now, for any track $\rho$, $\rho_{ds}$ features at most $|W|$ subsequences associated with distinct clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots$, and thus, if the following upper bound to the length of $\rho$ is exceeded, then there is at least one pair of $k$-indistinguishable occurrences of some descriptor element in $\rho_{ds}$: $|\rho| \leq 1 + (|\mathcal{C}_1|+1)^{k+2} + (|\mathcal{C}_2|+1)^{k+2} + \cdots + (|\mathcal{C}_s|+1)^{k+2} + |W|$, where $s \leq |W|$, and the last addend is to count occurrences of Type-1 descriptor elements. Since clusters are disjoint, their union is a subset of $DElm(\rho_{ds})$, and $|DElm(\rho_{ds})| \leq 1 + |W|^2$, we get:

$$\begin{aligned}|\rho| \leq 1 + (|\mathcal{C}_1| + |\mathcal{C}_2| + \cdots + |\mathcal{C}_s| + |W|)^{k+2} + |W| &\leq 1 + (|DElm(\rho_{ds})| + |W|)^{k+2} + |W| \\ &\leq 1 + (1 + |W|^2 + |W|)^{k+2} + |W| \leq 1 + (1+|W|)^{2k+4} + |W|.\end{aligned}$$

Analogously, by using the alternative bound to $\varepsilon(|\mathcal{C}|, k+3)$, we have that

$$\begin{aligned}|\rho| \leq 1 + (k+3)^{|\mathcal{C}_1|} + (k+3)^{|\mathcal{C}_2|} + \cdots + (k+3)^{|\mathcal{C}_s|} + |W| &\leq 1 + (k+3)^{|\mathcal{C}_1|+|\mathcal{C}_2|+\cdots+|\mathcal{C}_s|} + |W| \\ &\leq 1 + (k+3)^{|DElm(\rho_{ds})|} + |W| \leq 1 + (k+3)^{|W|^2+1} + |W|.\end{aligned}$$

The upper bound for $|\rho|$ is then the least of the two given upper bounds:

$$\tau(|W|, k) = \min\left\{1 + (1+|W|)^{2k+4} + |W|, 1 + (k+3)^{|W|^2+1} + |W|\right\}.$$

**Theorem 31.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure and $\rho$ be a track in $\mathrm{Trk}_{\mathcal{K}}$. If $|\rho| > \tau(|W|, k)$, then there exists another track in $\mathrm{Trk}_{\mathcal{K}}$, whose length is less than or equal to $\tau(|W|, k)$, associated with the same $B_k$-descriptor as $\rho$.*

*Proof (sketch).* If $|\rho| > \tau(|W|, k)$, then there exists (at least) a subsequence of $\rho_{ds}$, associated with some cluster $\mathcal{C}$, which contains (at least) a pair of $k$-indistinguishable occurrences of some descriptor element $d \in \mathcal{C}$, say $\rho_{ds}(i)$ and $\rho_{ds}(j)$, with $j < i$. By Theorem 25, the two tracks $\tilde{\rho}_1 = \rho(0, j+1)$ and $\tilde{\rho}_2 = \rho(0, i+1)$ have the same $B_k$-descriptor. Now, let us rewrite the track $\rho$ as the concatenation $\tilde{\rho}_2 \cdot \overline{\rho}$ for some $\overline{\rho}$. By Proposition 13, the tracks $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ and $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ are associated with the same $B_k$-descriptor. Since $\mathrm{lst}(\tilde{\rho}_1) = \mathrm{lst}(\tilde{\rho}_2)$ ($\rho_{ds}(j)$ and $\rho_{ds}(i)$ are occurrences of the same descriptor element $d$), $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ is a track of $\mathcal{K}$ shorter than $\rho$. If $|\rho'| \leq \tau(|W|, k)$, we have proved the thesis; otherwise, we can iterate the process by applying the above contraction to $\rho'$. □

Theorem 31 allows us to define a termination criterion to bound the depth of the unravelling of a finite Kripke structure ($(k \geq 1)$-*termination criterion*), while searching for track representatives for witnessed $B_k$-descriptors: *for any $k \geq 1$, to get a track representative for every $B_k$-descriptor, with initial state $v$, and witnessed in a finite Kripke structure with set of states $W$, we can avoid taking into consideration tracks longer than $\tau(|W|, k)$ while exploring the unravelling of the structure from $v$.*

Thanks to the above results, we are now ready to define a model checking algorithm for $\mathsf{A\overline{A}B\overline{B}E}$ formulas. First, we introduce the unravelling Algorithm 1, which explores the unravelling of the input Kripke structure $\mathcal{K}$ to find track representatives for all witnessed $B_k$-descriptors. It features two modalities, *forward mode* (which is active when its fourth parameter, direction, is FORW) and *backward mode* (active when the parameter direction is BACKW), in which the unravelling of $\mathcal{K}$ is visited following the direction of edges and against their direction (that is equivalent to visiting the

**Algorithm 1** `Unrav`$(\mathcal{K}, v, k, \text{direction})$

1: **if** direction = FORW **then**
2: Unravel $\mathcal{K}$ starting from $v$ according to $\ll$ ◁ *"$\ll$" is an arbitrary order of the nodes of $\mathcal{K}$*
3: For every new node *of the unravelling* met during the visit, return the track $\rho$ from $v$ to the current node only if:
4: **if** $k = 0$ **then**
5: Apply the 0-termination criterion
6: **else**
7: **if** The last descriptor element $d$ of (the descriptor sequence of) the current track $\rho$ is $k$-indistinguishable from a previous occurrence of $d$ **then**
8: *skip* $\rho$ and backtrack to $\rho(0, |\rho| - 2) \cdot \overline{v}$, where $\overline{v}$ is the minimum state (w.r.t. $\ll$), greater than $\rho(|\rho| - 1)$, such that $(\rho(|\rho| - 2), \overline{v})$ is an edge of $\mathcal{K}$.
9: **else if** direction = BACKW **then**
10: Unravel $\overline{\mathcal{K}}$ starting from $v$ according to $\ll$ ◁ *$\overline{\mathcal{K}}$ is $\mathcal{K}$ with transposed edges*
11: For every new node *of the unravelling* met during the visit, consider the track $\rho$ *from the current node to* $v$, and recalculate descriptor element indistinguishability from scratch (left to right); return the track only if:
12: **if** $k = 0$ **then**
13: Apply the 0-termination criterion
14: **else**
15: **if** There exist two $k$-indistinguishable occurrences of a descriptor element $d$ in (the descriptor sequence of) the current track $\rho$ **then**
16: *skip* $\rho$
17: Do not visit tracks of length greater than $\tau(|W|, k)$

transposed graph $\overline{\mathcal{K}}$ of $\mathcal{K}$), respectively. In both cases, if there exist $k$-indistinguishable occurrences of a descriptor element in $\rho_{ds}$, the track $\rho$ is never returned.

In the *forward mode* (which will be used to deal with $\langle \mathrm{A} \rangle$ and $\langle \overline{\mathrm{B}} \rangle$ modalities), the direction of track exploration and that of indistinguishability checking are the same, so we can stop extending a track as soon as the first pair of $k$-indistinguishable occurrences of a descriptor element is found in the descriptor sequence, suggesting an easy termination criterion for stopping the unravelling of tracks. In the *backward mode* (used in the case of $\langle \overline{\mathrm{A}} \rangle$ and $\langle \overline{\mathrm{E}} \rangle$ modalities), such a straightforward criterion cannot be adopted, because tracks are explored right to left (the opposite direction with respect to edges of the Kripke structure), while the indistinguishability relation over descriptor elements is computed left to right. In general, changing the prefix of a considered track requires recomputing from scratch the descriptor sequence and the indistinguishability relation over descriptor elements. In particular, $k$-indistinguishable occurrences of descriptor elements can be detected in the middle of a subsequence, and not necessarily at the end. In this latter case, however, the upper bound $\tau(|W|, k)$ on the maximum depth of the unravelling ensures the termination of the algorithm (line 17).

The next theorem proves soundness and completeness of Algorithm 1 for the forward mode. The proof for the backward one is quite similar, and thus omitted.

**Theorem 32.** *Let $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ be a finite Kripke structure, $v \in W$, and $k \in \mathbb{N}$. For every track $\rho$ of $\mathcal{K}$, with $\mathrm{fst}(\rho) = v$ and $|\rho| \geq 2$, the unravelling Algorithm 1 returns a track $\rho'$ of $\mathcal{K}$, with $\mathrm{fst}(\rho') = v$, such that $\rho$ and $\rho'$ are associated with the same $B_k$-descriptor and $|\rho'| \leq \tau(|W|, k)$.*

*Proof.* If $k = 0$ the thesis follows immediately by the 0-termination criterion. So let us assume

$k \geq 1$. The proof is by induction on $\ell = |\rho|$.

(Case $\ell = 2$) In this case, $\rho_{ds} = (\mathrm{fst}(\rho), \emptyset, \mathrm{lst}(\rho))$, and the only descriptor element of the sequence is Type-1. Thus, $\rho$ itself is returned by the algorithm.

(Case $\ell > 2$) If in $\rho_{ds}$ there are no pairs of $k$-indistinguishable occurrences of some descriptor element, the termination criterion of Algorithm 1 can never be applied. Thus, $\rho$ itself is returned (as soon as it is visited) and its length is at most $\tau(|W|, k)$.

Otherwise, the descriptor sequence of any track $\rho$ can be split into 3 parts: $\rho_{ds} = \rho_{ds1} \cdot \rho_{ds2} \cdot \rho_{ds3}$, where $\rho_{ds1}$ ends with a Type-1 descriptor element and it does not contain pairs of $k$-indistinguishable occurrences of any descriptor element; $\rho_{ds2}$ is a subsequence associated with a cluster $\mathcal{C}$ of (Type-2) descriptor elements with at least a pair of $k$-indistinguishable occurrences of descriptor elements; $\rho_{ds3}$ (if it is not the empty sequence) begins with a Type-1 descriptor element. This amounts to say that $\rho_{ds2}$ is the "leftmost" subsequence of $\rho_{ds}$ consisting of elements of a cluster $\mathcal{C}$, with at least a pair of $k$-indistinguishable occurrences of some descriptor element.

Therefore, there are two indexes $i, j$, with $j < i$, such that $\rho_{ds2}(j)$ and $\rho_{ds2}(i)$ are two $k$-indistinguishable occurrences of some $d \in \mathcal{C}$ in $\rho_{ds}$. By Proposition 24, there exists a pair of indexes $i', j'$, with $j' < i'$, such that $\rho_{ds2}(j')$ and $\rho_{ds2}(i')$ are two *consecutive* $k$-indistinguishable occurrences of $d$ (by consecutive we mean that, for all $t \in [j'+1, i'-1]$, $\rho_{ds2}(t) \neq d$). If there are many such pairs (even for different elements in $\mathcal{C}$), let us consider the one with the lower index $i'$ (namely, precisely the pair which is found earlier by the unravelling algorithm). By Theorem 25, the two tracks associated with $\rho_{ds1} \cdot \rho_{ds2}(0, j')$ and $\rho_{ds1} \cdot \rho_{ds2}(0, i')$, say $\tilde{\rho}_1$ and $\tilde{\rho}_2$ respectively, have the same $B_k$-descriptor. Then, by Proposition 13, the tracks $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ (for some $\overline{\rho}$) and $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$ have the same $B_k$-descriptor.

Algorithm 1 does not return $\tilde{\rho}_2$ and, due to the backtrack step, neither $\rho = \tilde{\rho}_2 \cdot \overline{\rho}$ is returned. But since $\mathrm{lst}(\tilde{\rho}_1) = \mathrm{lst}(\tilde{\rho}_2)$ ($\rho_{ds2}(j')$ and $\rho_{ds2}(i')$ are occurrences of the same descriptor element), the unravelling of $\mathcal{K}$ features $\rho' = \tilde{\rho}_1 \cdot \overline{\rho}$, as well. Now, by induction hypothesis, a track $\rho''$ of $\mathcal{K}$ is returned, such that $\rho'$ and $\rho''$ have the same $B_k$-descriptor, and $|\rho''| \leq \tau(|W|, k)$. $\rho$ has in turn the same $B_k$-descriptor as $\rho''$. □

The above proof shows how a "contracted variant" of a track $\rho$ is (indirectly) computed by Algorithm 1. As an example, $\rho' = v_0v_1v_2v_3v_3v_2v_3v_3v_2v_3v_2v_3v_2v_1v_3v_2v_3v_2v_1v_2v_1v_3v_2$ is returned by Algorithm 1 in place of the track $\rho$ of Example 4, and it can be checked that $\rho'_{ds}$ does not contain any pair of 3-indistinguishable occurrences of a descriptor element and that $\rho$ and $\rho'$ have the same $B_3$-descriptor.

Algorithm 1 can be used to define the model checking procedure $\texttt{ModCheck}(\mathcal{K}, \psi)$ (Algorithm 2). $\texttt{ModCheck}(\mathcal{K}, \psi)$ exploits the procedure $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho})$ (Algorithm 3), which checks a formula $\psi$ of B-nesting depth $k$ against a track $\tilde{\rho}$ of the Kripke structure $\mathcal{K}$. $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho})$ basically calls itself recursively on the subformulas of $\psi$, and uses the unravelling Algorithm 1 to deal with $\langle \mathrm{A} \rangle$, $\langle \overline{\mathrm{A}} \rangle$, $\langle \overline{\mathrm{B}} \rangle$, and $\langle \overline{\mathrm{E}} \rangle$ modalities. Soundness and completeness of these two procedures are stated by Lemma 33 and Theorem 34 below, whose proofs can be found in Appendix A.3 and Appendix A.4, respectively.

**Lemma 33.** *Let $\psi$ be an $\mathsf{A\overline{A}B\overline{B}E}$ formula with $\mathrm{Nest}_{\mathrm{B}}(\psi) = k$, $\mathcal{K}$ be a finite Kripke structure, and $\tilde{\rho}$ be a track in $\mathrm{Trk}_{\mathcal{K}}$. It holds that $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$ if and only if $\mathcal{K}, \tilde{\rho} \models \psi$.*

**Theorem 34.** *Let $\psi$ be an $\mathsf{A\overline{A}B\overline{B}E}$ formula and $\mathcal{K}$ be a finite Kripke structure. It holds that $\texttt{ModCheck}(\mathcal{K}, \psi) = 1$ if and only if $\mathcal{K} \models \psi$.*

**Algorithm 2** $\texttt{ModCheck}(\mathcal{K}, \psi)$

$k \leftarrow \mathrm{Nest_B}(\psi)$
$u \leftarrow New\,(\texttt{Unrav}(\mathcal{K}, w_0, k, \mathrm{FORW}))$ ◁ *$w_0$ is the initial state of $\mathcal{K}$*
**while** $u$.hasMoreTracks() **do**
  $\tilde{\rho} \leftarrow u$.getNextTrack()
  **if** $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 0$ **then**
    **return** 0: "$\mathcal{K}, \tilde{\rho} \not\models \psi$"
**return** 1: "$\mathcal{K} \models \psi$"

**Algorithm 3** $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho})$

**if** $\psi = \top$ **then**
  **return** 1
**else if** $\psi = \bot$ **then**
  **return** 0
**else if** $\psi = p \in \mathcal{AP}$ **then**
  **if** $p \in \bigcap_{s \in \mathrm{states}(\tilde{\rho})} \mu(s)$ **then**
    **return** 1 **else return** 0
**else if** $\psi = \neg\varphi$ **then**
  **return** $1 - \texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho})$
**else if** $\psi = \varphi_1 \wedge \varphi_2$ **then**
  **if** $\texttt{Check}(\mathcal{K}, k, \varphi_1, \tilde{\rho}) = 0$ **then**
    **return** 0
  **else**
    **return** $\texttt{Check}(\mathcal{K}, k, \varphi_2, \tilde{\rho})$
**else if** $\psi = \langle \mathrm{A} \rangle \varphi$ **then**
  $u \leftarrow New\,(\texttt{Unrav}(\mathcal{K}, \mathrm{lst}(\tilde{\rho}), k, \mathrm{FORW}))$
  **while** $u$.hasMoreTracks() **do**
    $\rho \leftarrow u$.getNextTrack()
    **if** $\texttt{Check}(\mathcal{K}, k, \varphi, \rho) = 1$ **then**
      **return** 1
  **return** 0
**else if** $\psi = \langle \overline{\mathrm{A}} \rangle \varphi$ **then**
  $u \leftarrow New\,(\texttt{Unrav}(\mathcal{K}, \mathrm{fst}(\tilde{\rho}), k, \mathrm{BACKW}))$
  **while** $u$.hasMoreTracks() **do**
    $\rho \leftarrow u$.getNextTrack()
    **if** $\texttt{Check}(\mathcal{K}, k, \varphi, \rho) = 1$ **then**
      **return** 1
  **return** 0
**else if** $\psi = \langle \mathrm{B} \rangle \varphi$ **then**
  **for** each $\overline{\rho}$ prefix of $\tilde{\rho}$ **do**
    **if** $\texttt{Check}(\mathcal{K}, k-1, \varphi, \overline{\rho}) = 1$ **then**
      **return** 1
  **return** 0
**else if** $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$ **then**
  **for** each $v \in W$ s.t. $(\mathrm{lst}(\tilde{\rho}), v) \in \delta$ **do**
    **if** $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot v) = 1$ **then**
      **return** 1
    $u \leftarrow New\,(\texttt{Unrav}(\mathcal{K}, v, k, \mathrm{FORW}))$
    **while** $u$.hasMoreTracks() **do**
      $\rho \leftarrow u$.getNextTrack()
      **if** $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$ **then**
        **return** 1
  **return** 0
**else if** $\psi = \langle \overline{\mathrm{E}} \rangle \varphi$ **then**
  **for** each $v \in W$ s.t. $(v, \mathrm{fst}(\tilde{\rho})) \in \delta$ **do**
    **if** $\texttt{Check}(\mathcal{K}, k, \varphi, v \cdot \tilde{\rho}) = 1$ **then**
      **return** 1
    $u \leftarrow New\,(\texttt{Unrav}(\mathcal{K}, v, k, \mathrm{BACKW}))$
    **while** $u$.hasMoreTracks() **do**
      $\rho \leftarrow u$.getNextTrack()
      **if** $\texttt{Check}(\mathcal{K}, k, \varphi, \rho \cdot \tilde{\rho}) = 1$ **then**
        **return** 1
  **return** 0

The model checking algorithm `ModCheck` requires *exponential working space*, as it uses an instance of the unravelling algorithm and some additional space for a track $\tilde{\rho}$. Analogously, every recursive call to `Check` (possibly) needs an instance of the unravelling algorithm and space for a track. There are at most $|\psi|$ jointly active calls to `Check` (plus one to `ModCheck`), thus the maximum space needed by the considered algorithms is $(|\psi|+1)\cdot O(|W|+\mathrm{Nest}_{\mathrm{B}}(\psi))\cdot\tau(|W|,\mathrm{Nest}_{\mathrm{B}}(\psi))$ bits overall, where $\tau(|W|,\mathrm{Nest}_{\mathrm{B}}(\psi))$ is the maximum length of track representatives, and $O(|W|+\mathrm{Nest}_{\mathrm{B}}(\psi))$ bits are needed to represent a state of $\mathcal{K}$, a descriptor element, and a counter for $k$-indistinguishability.

In conclusion, we have proved that the model checking problem for formulas of the HS fragment $\mathsf{A\overline{A}B\overline{BE}}$ over finite Kripke structures is in EXPSPACE. As a particular case, formulas $\psi$ of the fragment $\mathsf{A\overline{ABE}}$ can be checked in *polynomial working space* by `ModCheck`, as its formulas do not feature $\langle\mathrm{B}\rangle$ modality (hence $\mathrm{Nest}_{\mathrm{B}}(\psi)=0$). Thus, the model checking problem for $\mathsf{A\overline{ABE}}$ is in PSPACE. In the next section, we prove that it is actually PSPACE-complete. As a direct consequence, $\mathsf{A\overline{A}B\overline{BE}}$ turns out to be PSPACE-hard.

The next theorem proves that the model checking problem for $\mathsf{A\overline{A}B\overline{BE}}$ is NEXP-hard if a *succinct* encoding of formulas is adopted (the proof is given in Appendix A.5).

**Theorem 39.** *The model checking problem for succinctly encoded formulas of* $\mathsf{A\overline{A}B\overline{BE}}$ *over finite Kripke structures is NEXP-hard (under polynomial-time reductions).*

## 6. The fragment $\mathsf{A\overline{ABE}}$

In this section, we prove that the model checking algorithm described in the previous section, applied to $\mathsf{A\overline{ABE}}$ formulas, is optimal by showing that model checking for $\mathsf{A\overline{B}}$ is a PSPACE-hard problem (Theorem 36). PSPACE-completeness of $\mathsf{A\overline{ABE}}$ (and $\mathsf{A\overline{B}}$) immediately follows. As a by-product, model checking for $\mathsf{A\overline{A}B\overline{BE}}$ is PSPACE-hard as well.

Before proving Theorem 36, we give an example showing that the three HS fragments $\mathsf{A\overline{ABE}}$, $\forall\mathsf{A\overline{A}BE}$, and $\mathsf{A\overline{A}}$, on which we focus in this (and the next) section, are expressive enough to capture meaningful properties of state-transition systems.

**Example 5.** *Let* $\mathcal{K}=(\mathcal{AP},W,\delta,\mu,w_0)$, *with* $\mathcal{AP}=\{r_0,r_1,e_0,e_1,x_0\}$, *be the Kripke structure of Figure 8, that models the interactions between a scheduler* $\mathcal{S}$ *and two processes,* $\mathcal{P}_0$ *and* $\mathcal{P}_1$, *which possibly ask for a shared resource. At the initial state* $w_0$, $\mathcal{S}$ *has not received any request from the processes yet, while in* $w_1$ *(resp.,* $w_2$*) only* $\mathcal{P}_0$ *(resp.,* $\mathcal{P}_1$*) has sent a request, and thus* $r_0$ *(resp.,* $r_1$*) holds. As long as at most one process has issued a request,* $\mathcal{S}$ *is not forced to allocate the resource (*$w_1$ *and* $w_2$ *have self loops). At state* $w_3$, *both* $\mathcal{P}_0$ *and* $\mathcal{P}_1$ *are waiting for the shared resource (both* $r_0$ *and* $r_1$ *hold). State* $w_3$ *has transitions only towards* $w_4$, $w_6$, *and* $w_8$. *At state* $w_4$ *(resp.,* $w_6$*)* $\mathcal{P}_1$ *(resp.,* $\mathcal{P}_0$*) can access the resource and* $e_1$ *(resp.,* $e_0$*) holds in the interval* $w_4w_5$ *(resp.,* $w_6w_7$*). In addition, a faulty transition may be taken from* $w_3$ *leading to states* $w_8$ *and* $w_9$ *where both* $\mathcal{P}_0$ *and* $\mathcal{P}_1$ *use the resource (both* $e_0$ *and* $e_1$ *hold in the interval* $w_8w_9$*). Finally, from* $w_5$, $w_7$, *and* $w_9$ *the system can only move to* $w_0$, *where* $\mathcal{S}$ *waits for new requests from* $\mathcal{P}_0$ *and* $\mathcal{P}_1$.

*Let* $\mathcal{P}$ *be the set* $\{r_0,r_1,e_0,e_1\}$ *and let* $x_0$ *be an auxiliary proposition letter labelling the states* $w_0$, $w_1$, $w_6$, *and* $w_7$, *where* $\mathcal{S}$ *and* $\mathcal{P}_0$, *but not* $\mathcal{P}_1$, *are active.*

*We now give some examples of formulas in the fragments* $\mathsf{A\overline{ABE}}$, $\forall\mathsf{A\overline{A}BE}$, *and* $\mathsf{A\overline{A}}$ *that encode requirements for* $\mathcal{K}$. *As in Example 3, we force the validity of the considered property over all legal computation sub-intervals by using the modality* $[E]$, *or alternatively the modality* $[A]$ *(any computation sub-interval occurs after at least one initial track).*

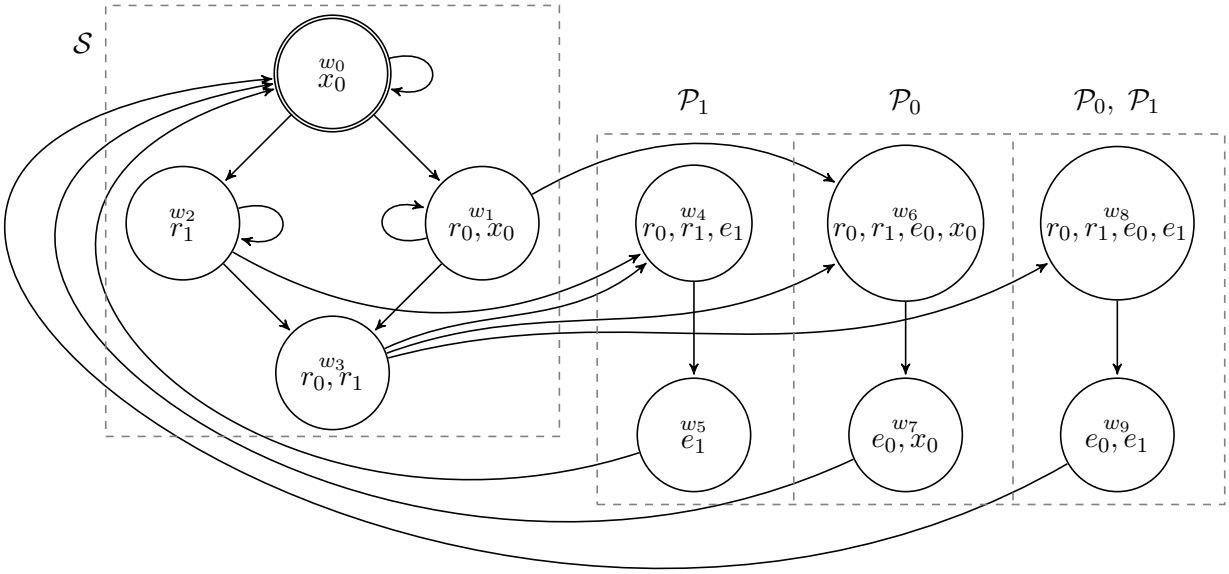



Figure 8: A simple state-transition system.

*It can be checked that* $\mathcal{K} \not\models [E]\neg(e_0 \wedge e_1)$ *(the formula is in* $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}$*), i.e., mutual exclusion is not guaranteed, as the faulty transition leading to* $w_8$ *may be taken at* $w_3$*, and then both* $\mathcal{P}_0$ *and* $\mathcal{P}_1$ *access the resource in the interval* $w_8w_9$ *where* $e_0 \wedge e_1$ *holds.*

*On the contrary, it holds that* $\mathcal{K} \models [A]\big(r_0 \rightarrow \langle \mathrm{A}\rangle\, e_0 \vee \langle \mathrm{A}\rangle\, \langle \mathrm{A}\rangle\, e_0\big)$ *(the formula is in* $\mathsf{A}\overline{\mathsf{A}}$ *and* $\mathsf{A}\overline{\mathsf{A}\mathsf{B}\mathsf{E}}$*). Such a formula expresses the following reachability property: if* $r_0$ *holds over some interval, then it is always possible to reach an interval where* $e_0$ *holds. Obviously, this does not mean that all possible computations will necessarily lead to such an interval, but that the system is* never *trapped in a state from which it is no more possible to satisfy requests from* $\mathcal{P}_0$*.*

*It also holds that* $\mathcal{K} \models [A]\big(r_0 \wedge r_1 \rightarrow [A](e_0 \vee e_1 \vee \bigwedge_{p\in\mathcal{P}} \neg p)\big)$ *(in* $\mathsf{A}\overline{\mathsf{A}}$ *and* $\mathsf{A}\overline{\mathsf{A}\mathsf{B}\mathsf{E}}$*). Indeed, if both processes send a request (state* $w_3$*), then* $\mathcal{S}$ *immediately allocates the resource. In detail, if* $r_0 \wedge r_1$ *holds over some tracks (the only possible intervals are* $w_3w_4$*,* $w_3w_6$*, and* $w_3w_8$*), then in any possible subsequent interval of length 2* $e_0 \vee e_1$ *holds, that is,* $\mathcal{P}_0$ *or* $\mathcal{P}_1$ *are executed, or, considering tracks longer than 2,* $\bigwedge_{p\in\mathcal{P}} \neg p$ *holds. On the contrary, if only one process asks for the resource, then* $\mathcal{S}$ *can arbitrarily delay the allocation, and therefore* $\mathcal{K} \not\models [A]\big(r_0 \rightarrow [A](e_0 \vee \bigwedge_{p\in\mathcal{P}} \neg p)\big)$*.*

*Finally, it holds that* $\mathcal{K} \models x_0 \rightarrow \langle \overline{\mathrm{B}}\rangle\, x_0$ *(in* $\mathsf{A}\overline{\mathsf{A}\mathsf{B}\mathsf{E}}$*), that is, any initial track satisfying* $x_0$ *(any such track involves states* $w_0$*,* $w_1$*,* $w_6$*, and* $w_7$ *only) can be extended to the right in such a way that the resulting track still satisfies* $x_0$*. This amounts to say that there exists a computation in which* $\mathcal{P}_1$ *starves. Note that* $\mathcal{S}$ *and* $\mathcal{P}_0$ *can continuously interact without waiting for* $\mathcal{P}_1$*. This is the case, for instance, when* $\mathcal{P}_1$ *is not asking for the shared resource at all.*

Now, in order to prove Theorem 36, we provide a reduction from the QBF problem (i.e., the problem of determining the truth of a *fully-quantified* Boolean formula in *prenex normal form*)—which is known to be PSPACE-complete (see, for example, [33])—to the model checking problem for $\mathsf{A}\overline{\mathsf{B}}$ formulas over finite Kripke structures.

We consider a quantified Boolean formula $\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots, x_1)$ where $Q_i \in \{\exists, \forall\}$ for all $i = 1, \cdots, n$, and $\phi(x_n, x_{n-1}, \cdots, x_1)$ is a quantifier-free Boolean formula. Let $Var = \{x_n, \ldots, x_1\}$ be the set of variables of $\psi$. We define the Kripke structure $\mathcal{K}_{QBF}^{Var}$, whose

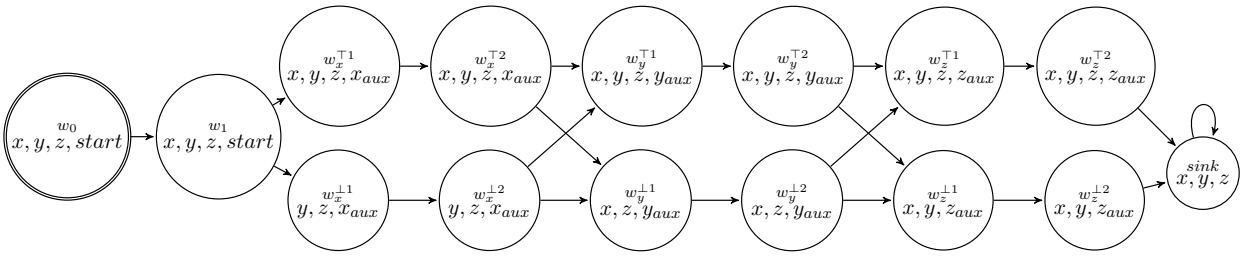



Figure 9: Kripke structure $\mathcal{K}_{QBF}^{x,y,z}$ associated with a quantified Boolean formula with variables $x$, $y$, $z$.

initial tracks represent all the possible assignments to the variables of $Var$. For each $x \in Var$, $\mathcal{K}_{QBF}^{Var}$ features four states, $w_x^{\top 1}$, $w_x^{\top 2}$, $w_x^{\perp 1}$, and $w_x^{\perp 2}$: the first two represent a $\top$ truth assignment to $x$ and the last two a $\perp$ one. $\mathcal{K}_{QBF}^{Var} = (\mathcal{AP}, W, \delta, \mu, w_0)$ is formally defined as follows:

- $\mathcal{AP} = Var \cup \{start\} \cup \{x_{i\,aux} \mid 1 \leq i \leq n\}$;
- $W = \{w_{x_i}^{\ell} \mid 1 \leq i \leq n, \ell \in \{\perp_1, \perp_2, \top_1, \top_2\}\} \cup \{w_0, w_1, sink\}$;
- if $n = 0$, $\delta = \{(w_0, w_1), (w_1, sink), (sink, sink)\}$;
  if $n > 0$, $\delta = \{(w_0, w_1), (w_1, w_{x_n}^{\top_1}), (w_1, w_{x_n}^{\perp_1})\} \cup \{(w_{x_i}^{\top_1}, w_{x_i}^{\top_2}), (w_{x_i}^{\perp_1}, w_{x_i}^{\perp_2}) \mid 1 \leq i \leq n\} \cup \{(w_{x_i}^{\ell}, w_{x_{i-1}}^{m}) \mid \ell \in \{\perp_2, \top_2\}, m \in \{\perp_1, \top_1\}, 2 \leq i \leq n\} \cup \{(w_{x_1}^{\top_2}, sink), (w_{x_1}^{\perp_2}, sink)\} \cup \{(sink, sink)\}$.
- $\mu(w_0) = \mu(w_1) = Var \cup \{start\}$;
  $\mu(w_{x_i}^{\ell}) = Var \cup \{x_{i\,aux}\}$, for $1 \leq i \leq n$ and $\ell \in \{\top_1, \top_2\}$;
  $\mu(w_{x_i}^{\ell}) = (Var \setminus \{x_i\}) \cup \{x_{i\,aux}\}$, for $1 \leq i \leq n$ and $\ell \in \{\perp_1, \perp_2\}$;
  $\mu(sink) = Var$.

An example of such a Kripke structure, for $Var = \{x, y, z\}$, is given in Figure 9.
From $\psi$, we obtain the $\mathsf{A}\overline{\mathsf{B}}$ formula $\xi = start \rightarrow \xi_n$, where

$$\xi_i = \begin{cases} \phi(x_n, x_{n-1}, \cdots, x_1) & i = 0 \\ \langle \overline{\mathrm{B}} \rangle \left( (\langle \mathrm{A} \rangle\, x_{i\,aux}) \wedge \xi_{i-1} \right) & i > 0 \wedge Q_i = \exists \\ [\overline{B}] \left( (\langle \mathrm{A} \rangle\, x_{i\,aux}) \rightarrow \xi_{i-1} \right) & i > 0 \wedge Q_i = \forall \end{cases}$$

Both $\mathcal{K}_{QBF}^{Var}$ and $\xi$ can be built by using logarithmic working space. We will show (proof of Theorem 36) that $\psi$ is true if and only if $\mathcal{K}_{QBF}^{Var} \models \xi$.

As a preliminary step, we introduce some technical definitions. Given a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and an $\mathsf{A}\overline{\mathsf{B}}$ formula $\psi$, we denote by $p\ell(\psi)$ the set of proposition letters occurring in $\psi$ and by $\mathcal{K}_{|p\ell(\psi)}$ the structure obtained from $\mathcal{K}$ by restricting the labelling of each state to $p\ell(\psi)$, namely, the Kripke structure $(\overline{\mathcal{AP}}, W, \delta, \overline{\mu}, w_0)$, where $\overline{\mathcal{AP}} = \mathcal{AP} \cap p\ell(\psi)$ and $\overline{\mu}(w) = \mu(w) \cap p\ell(\psi)$, for all $w \in W$. Moreover, for $v \in W$, we denote by $reach(\mathcal{K}, v)$ the subgraph of $\mathcal{K}$, with $v$ as its initial state, consisting of all and only the states which are reachable from $v$, namely, the Kripke structure $(\mathcal{AP}, W', \delta', \mu', v)$, where $W' = \{w \in W \mid$ there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ with $\mathrm{fst}(\rho) = v$ and $\mathrm{lst}(\rho) = w\}$, $\delta' = \delta \cap (W' \times W')$, and $\mu'(w) = \mu(w)$, for all $w \in W'$.

As usual, two Kripke structures $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w_0')$ are said to be *isomorphic* ($\mathcal{K} \sim \mathcal{K}'$ for short) if and only if there is a *bijection* $f : W \mapsto W'$ such that

$(i)$ $f(w_0) = w_0'$; $(ii)$ for all $u, v \in W$, $(u, v) \in \delta$ if and only if $(f(u), f(v)) \in \delta'$; $(iii)$ for all $v \in W$, $\mu(v) = \mu'(f(v))$.

Finally, if $\mathcal{A}_\mathcal{K} = (\mathcal{AP}, \mathbb{I}, A_\mathbb{I}, B_\mathbb{I}, E_\mathbb{I}, \sigma)$ is the abstract interval model induced by a Kripke structure $\mathcal{K}$ and $\rho \in \mathrm{Trk}_\mathcal{K}$, we denote $\sigma(\rho)$ by $\mathcal{L}(\mathcal{K}, \rho)$.

Let $\mathcal{K}$ and $\mathcal{K}'$ be two Kripke structures. The following lemma states that, for any $\mathsf{A\overline{B}}$ formula $\psi$, if the same set of proposition letters, restricted to $p\ell(\psi)$, holds over two tracks $\rho \in \mathrm{Trk}_\mathcal{K}$ and $\rho' \in \mathrm{Trk}_{\mathcal{K}'}$, and the subgraphs consisting of the states reachable from, respectively, $\mathrm{lst}(\rho)$ and $\mathrm{lst}(\rho')$ are isomorphic, then $\rho$ and $\rho'$ are equivalent with respect to $\psi$.

**Lemma 35.** *Given an $\mathsf{A\overline{B}}$ formula $\psi$, two Kripke structures $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and $\mathcal{K}' = (\mathcal{AP}', W', \delta', \mu', w_0')$, and two tracks $\rho \in \mathrm{Trk}_\mathcal{K}$ and $\rho' \in \mathrm{Trk}_{\mathcal{K}'}$ such that*

$$\mathcal{L}(\mathcal{K}_{|p\ell(\psi)}, \rho) = \mathcal{L}(\mathcal{K}'_{|p\ell(\psi)}, \rho') \quad \textit{and} \quad \mathit{reach}(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho)) \sim \mathit{reach}(\mathcal{K}'_{|p\ell(\psi)}, \mathrm{lst}(\rho')),$$

*it holds that $\mathcal{K}, \rho \models \psi \iff \mathcal{K}', \rho' \models \psi$.*

The proof of this lemma can be found in Appendix A.6.

**Theorem 36.** *The model checking problem for $\mathsf{A\overline{B}}$ formulas over finite Kripke structures is PSPACE-hard (under LOGSPACE reductions).*

*Proof.* We prove that the quantified Boolean formula $\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots, x_1)$ is true if and only if $\mathcal{K}_{QBF}^{x_n, \cdots, x_1} \models \xi$ by induction on the number of variables $n \geq 0$ of $\psi$. In the following, $\phi(x_n, x_{n-1}, \cdots, x_1)\{x_i/\upsilon\}$, with $\upsilon \in \{\top, \bot\}$, denotes the formula obtained from $\phi(x_n, x_{n-1}, \cdots, x_1)$ by replacing all occurrences of $x_i$ by $\upsilon$. It is worth noticing that $\mathcal{K}_{QBF}^{x_n, x_{n-1}, \cdots, x_1}$ and $\mathcal{K}_{QBF}^{x_{n-1}, \cdots, x_1}$ are isomorphic when they are restricted to the states $w_{x_{n-1}}^{\top 1}$, $w_{x_{n-1}}^{\top 2}$, $w_{x_{n-1}}^{\bot 1}$, $w_{x_{n-1}}^{\bot 2}$, $\cdots$, $w_{x_1}^{\top 1}$, $w_{x_1}^{\top 2}$, $w_{x_1}^{\bot 1}$, $w_{x_1}^{\bot 2}$, *sink* (i.e., the leftmost part of both Kripke structures is omitted), and the labelling of states is suitably restricted accordingly. Note that only the track $w_0 w_1$ satisfies *start* and, for $i = n, \cdots, 1$, the proposition letter $x_{i\,aux}$ is satisfied by the two tracks $w_{x_i}^{\top 1} w_{x_i}^{\top 2}$ and $w_{x_i}^{\bot 1} w_{x_i}^{\bot 2}$ only.

(Case $n = 0$) $\psi$ equals $\phi$ and it has no variables. The states of $\mathcal{K}_{QBF}^{\emptyset}$ are $W = \{w_0, w_1, sink\}$ and $\xi = start \rightarrow \phi$.

Let us assume $\phi$ to be true. All initial tracks of length greater than 2 trivially satisfy $\xi$, as *start* does not hold on them. As for $w_0 w_1$, it is true that $\mathcal{K}_{QBF}^{\emptyset}, w_0 w_1 \models \phi$, since $\phi$ is true (its truth does not depend on the proposition letters that hold on $w_0 w_1$, because it has no variables). Thus $\mathcal{K}_{QBF}^{\emptyset} \models \xi$. Vice versa, if $\mathcal{K}_{QBF}^{\emptyset} \models \xi$, then in particular $\mathcal{K}_{QBF}^{\emptyset}, w_0 w_1 \models \phi$. But $\phi$ has no variables, hence it is true.

(Case $n \geq 1$) Let us consider the formula $\psi = Q_n x_n Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi(x_n, x_{n-1}, \cdots, x_1)$. We distinguish two cases, depending on whether $Q_n = \exists$ or $Q_n = \forall$, and for both we prove the two implications.

$\circ$ Case $Q_n = \exists$:

($\Rightarrow$) If the formula $\psi$ is true, then, by definition, there exists $\upsilon \in \{\top, \bot\}$ such that if we replace all occurrences of $x_n$ in $\phi(x_n, x_{n-1}, \cdots, x_1)$ by $\upsilon$, we get the formula $\phi'(x_{n-1}, \cdots, x_1) = \phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\upsilon\}$ such that $\psi' = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \phi'(x_{n-1}, \cdots, x_1)$ is a true quantified Boolean formula. By the inductive hypothesis $\mathcal{K}_{QBF}^{x_{n-1}, \cdots, x_1} \models \xi'$, where $\xi' = start \rightarrow \xi'_{n-1}$ is

obtained from $\psi'$ and $\xi'_{n-1} = \xi_{n-1}\{x_n/\upsilon\}$. It follows that $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}, w'_0w'_1 \models \xi'_{n-1}$, where $w'_0$ and $w'_1$ are the two "leftmost" states of $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}$ (corresponding to $w_0$ and $w_1$ of $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}$).

We now prove that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1} \models \xi$. Let us consider a generic initial track $\rho$ in $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}$. If it does not satisfy *start*, then it trivially holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, \rho \models \xi$. Otherwise $\rho = w_0w_1$, and we have to show that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1 \models \langle\overline{\mathrm{B}}\rangle((\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1})$ $(= \xi_n)$. If $\upsilon = \top$, we consider $w_0w_1w_{x_n}^{\top 1}$; otherwise, we consider $w_0w_1w_{x_n}^{\bot 1}$. In the first case (the other is symmetric), we must prove that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models (\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1}$. It trivially holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \langle \mathrm{A}\rangle\, x_{n\,aux}$. Hence, we only need to prove that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \xi_{n-1}$.

As we have shown, by the inductive hypothesis, it holds that $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}, w'_0w'_1 \models \xi'_{n-1}(= \xi_{n-1}\{x_n/\top\})$. Now, since

- $p\ell(\xi_{n-1}\{x_n/\top\}) = \{x_1, \cdots, x_{n-1}, x_{1\,aux}, \cdots, x_{n-1\,aux}\}$,
- $\mathcal{L}(\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w'_0w'_1) = \{x_{n-1}, \cdots, x_1\}$,
- $\mathcal{L}(\mathcal{K}_{QBF}^{x_n,\cdots,x_1}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}) = \{x_{n-1}, \cdots, x_1\}$, and
- $reach(\mathcal{K}_{QBF}^{x_n,\cdots,x_1}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w_{x_n}^{\top 2}) \sim reach(\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}{}_{|p\ell(\xi_{n-1}\{x_n/\top\})}, w'_1)$,

by Lemma 35, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models \xi'_{n-1}$. Hence, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models \xi_{n-1}$ as $x_n$ is in the labelling of the track $w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}$ and of any $\overline{\rho}$ such that $w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \in \mathrm{Pref}(\overline{\rho})$.

Now, if $n = 1$, then $\xi_{n-1} = \phi(x_n)$ and it holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \xi_{n-1}$. If $n > 1$, either $\xi_{n-1} = \langle\overline{\mathrm{B}}\rangle((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2})$ or $\xi_{n-1} = [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \rightarrow \xi_{n-2})$. In the first case, since $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models \langle\overline{\mathrm{B}}\rangle((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2})$, there are only two possibilities: $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}w_{x_{n-1}}^{\top 1} \models (\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2}$ or $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}w_{x_{n-1}}^{\bot 1} \models (\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2}$. In both cases, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \langle\overline{\mathrm{B}}\rangle((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \wedge \xi_{n-2})$.

Otherwise, it holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n-1\,aux}) \rightarrow \xi_{n-2})$. It follows that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}w_{x_{n-1}}^{\top 1} \models \xi_{n-2}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2}w_{x_{n-1}}^{\bot 1} \models \xi_{n-2}$. As a consequence, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models [\overline{B}]((\neg\, \langle \mathrm{A}\rangle\, x_{n-1\,aux}) \vee \xi_{n-2})$ $(= \xi_{n-1})$ (recall that the only successor of $w_{x_n}^{\top 1}$ in $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}$ is $w_{x_n}^{\top 2}$ and, in particular, $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models \neg\, \langle \mathrm{A}\rangle\, x_{n-1\,aux}$).

($\Leftarrow$) If $\mathcal{K}_{QBF}^{x_n,\cdots,x_1} \models \xi$, it holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1 \models \langle\overline{\mathrm{B}}\rangle(\langle \mathrm{A}\rangle\, x_{n\,aux} \wedge \xi_{n-1})$. Hence, either $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models (\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1}$ or $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\bot 1} \models (\langle \mathrm{A}\rangle\, x_{n\,aux}) \wedge \xi_{n-1}$. Let us consider the first case (the other is symmetric). It holds that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \xi_{n-1}\{x_n/\top\}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1}w_{x_n}^{\top 2} \models \xi_{n-1}\{x_n/\top\}$ (as before). By Lemma 35 we get that $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1}, w'_0w'_1 \models \xi_{n-1}\{x_n/\top\}(= \xi'_{n-1})$ and thus $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1} \models start \rightarrow \xi'_{n-1}$, namely, $\mathcal{K}_{QBF}^{x_{n-1},\cdots,x_1} \models \xi'$. By the inductive hypothesis, $\psi' = Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\top\}$ is true. Hence, $\psi = \exists x_nQ_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)$ is true.

$\circ$ Case $Q_n = \forall$:

($\Rightarrow$) Assume that both $\psi' = Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\top\}$ and $\psi'' = Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\bot\}$ are true quantified Boolean formulas. We show that $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1 \models [\overline{B}]((\langle \mathrm{A}\rangle\, x_{n\,aux}) \rightarrow \xi_{n-1})$. To this end, we prove that both $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\bot 1} \models \xi_{n-1}$. This can be shown exactly as in the $\exists$ case.

($\Leftarrow$) If $\mathcal{K}_{QBF}^{x_n,\cdots,x_1} \models \xi$, then $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1 \models [\overline{B}](\langle \mathrm{A}\rangle x_{n\,aux}) \rightarrow \xi_{n-1})$. Hence, both $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\top 1} \models \xi_{n-1}$ and $\mathcal{K}_{QBF}^{x_n,\cdots,x_1}, w_0w_1w_{x_n}^{\bot 1} \models \xi_{n-1}$. Reasoning as in the $\exists$ case and by applying the inductive hypothesis twice, we get that the quantified Boolean formulas $Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\top\}$ and $Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)\{x_n/\bot\}$ are true; thus $\forall x_n Q_{n-1}x_{n-1}\cdots Q_1x_1\phi(x_n, x_{n-1}, \cdots, x_1)$ is true. □

## 7. The fragment $\forall\mathsf{A\overline{A}BE}$

In this section, we introduce and study the complexity of the model checking problem for the universal fragment of $\mathsf{A\overline{A}BE}$, denoted by $\forall\mathsf{A\overline{A}BE}$. Its formulas are defined as follows:

$$\psi ::= \beta \mid \psi \wedge \psi \mid [A]\psi \mid [B]\psi \mid [E]\psi \mid [\overline{A}]\psi,$$

where $\beta$ is a pure propositional formula,

$$\beta ::= p \mid \beta \vee \beta \mid \beta \wedge \beta \mid \neg\beta \mid \bot \mid \top \text{ with } p \in \mathcal{AP}.$$

Formulas of $\forall\mathsf{A\overline{A}BE}$ can thus be constructed starting from pure propositional formulas (a fragment of HS that we denote by $\mathsf{Prop}$); subsequently, formulas with universal modalities $[A]$, $[B]$, $[E]$, and $[\overline{A}]$ can be combined only by conjunctions, but not by negations or disjunctions (which may occur in pure propositional formulas only).

We will prove that the model checking problem for $\forall\mathsf{A\overline{A}BE}$ formulas (as well as for $\mathsf{Prop}$) over finite Kripke structures is coNP-complete.

To start with, we need to introduce the (auxiliary) fragment $\exists\mathsf{A\overline{A}BE}$, which can be regarded as the "dual" of $\forall\mathsf{A\overline{A}BE}$. Its formulas are defined as:

$$\psi ::= \beta \mid \psi \vee \psi \mid \langle A\rangle\psi \mid \langle B\rangle\psi \mid \langle E\rangle\psi \mid \langle\overline{A}\rangle\psi.$$

$\exists\mathsf{A\overline{A}BE}$ formulas feature $\langle\mathrm{A}\rangle$, $\langle\mathrm{B}\rangle$, $\langle\mathrm{E}\rangle$, and $\langle\overline{\mathrm{A}}\rangle$ existential modalities; negation and conjunction symbols may occur only in pure propositional formulas. The intersection of $\forall\mathsf{A\overline{A}BE}$ and $\exists\mathsf{A\overline{A}BE}$ is precisely $\mathsf{Prop}$. The negation of any $\forall\mathsf{A\overline{A}BE}$ formula can be transformed into an equivalent $\exists\mathsf{A\overline{A}BE}$ formula (of at most double length), and vice versa, by using De Morgan's laws and the equivalences $[X]\psi \equiv \neg\langle X\rangle\neg\psi$ and $\neg\neg\psi \equiv \psi$.

In the following, we outline a *non-deterministic* algorithm to decide the model checking problem for a $\forall\mathsf{A\overline{A}BE}$ formula $\psi$ (Algorithm 5). As usual, the algorithm searches for a counterexample to $\psi$, that is, an initial track satisfying $\neg\psi$. Since, as we already pointed out, $\neg\psi$ is equivalent to a suitable formula $\psi'$ of the dual fragment $\exists\mathsf{A\overline{A}BE}$, the algorithm looks for an initial track satisfying $\psi'$.

Algorithm 5 makes use of *descriptor elements*: we remind that they are the labels of the nodes of $B_k$-descriptors. By Proposition 27, if a descriptor element $d$ is witnessed in $\mathcal{K}$, i.e., there exists some $\rho \in \mathrm{Trk}_{\mathcal{K}}$ associated with $d$, then there exists a track of length at most $2 + |W|^2$ associated with $d$. Thus, to generate a (all) witnessed descriptor element(s) with initial state $v$, we just need to non-deterministically visit the unravelling of $\mathcal{K}$ from $v$ up to depth $2 + |W|^2$. This property is fundamental for the completeness of the algorithm, and also for bounding the length of tracks we need to consider.

Before presenting Algorithm 5, we need to describe the non-deterministic auxiliary procedure `Check`$\exists$ (see Algorithm 4), which takes as input a Kripke structure $\mathcal{K}$, a formula $\psi$ of $\exists\mathsf{A\overline{A}BE}$, and

**Algorithm 4** $\texttt{Check}\exists(\mathcal{K}, \psi, (v_{in}, S, v_{fin}))$

**if** $\psi = \beta$ **then** ◁ *$\beta$ is a pure propositional formula*
  **if** $VAL(\beta, (v_{in}, S, v_{fin})) = \top$ **then**
    Yes **else** No
**else if** $\psi = \varphi_1 \vee \varphi_2$ **then**
  **Either**
    **return** $\texttt{Check}\exists(\mathcal{K}, \varphi_1, (v_{in}, S, v_{fin}))$
  **Or**
    **return** $\texttt{Check}\exists(\mathcal{K}, \varphi_2, (v_{in}, S, v_{fin}))$
  **EndOr**
**else if** $\psi = \langle \mathrm{A} \rangle \varphi$ **then**
  $(v_{fin}, S', v'_{fin}) \leftarrow \texttt{aDescrEl}(\mathcal{K}, v_{fin}, \text{FORW})$
  **return** $\texttt{Check}\exists(\mathcal{K}, \varphi, (v_{fin}, S', v'_{fin}))$
**else if** $\psi = \langle \overline{\mathrm{A}} \rangle \varphi$ **then**
  $(v'_{in}, S', v_{in}) \leftarrow \texttt{aDescrEl}(\mathcal{K}, v_{in}, \text{BACKW})$
  **return** $\texttt{Check}\exists(\mathcal{K}, \varphi, (v'_{in}, S', v_{in}))$
**else if** $\psi = \langle \mathrm{B} \rangle \varphi$ **then**
  $(v'_{in}, S', v'_{fin}) \leftarrow \texttt{aDescrEl}(\mathcal{K}, v_{in}, \text{FORW})$ ◁ $v'_{in} = v_{in}$
  **Either**
    **if** $(v'_{in}, S' \cup \{v'_{fin}\}, v_{fin}) = (v_{in}, S, v_{fin})$ and $(v'_{fin}, v_{fin})$ is an edge of $\mathcal{K}$ **then**
      **return** $\texttt{Check}\exists(\mathcal{K}, \varphi, (v'_{in}, S', v'_{fin}))$
    **else**
      No
  **Or**
    $(v''_{in}, S'', v''_{fin}) \leftarrow \texttt{aDescrEl}(\mathcal{K}, v''_{in}, \text{FORW})$, where $(v'_{fin}, v''_{in})$ is an edge of $\mathcal{K}$ chosen non-deterministically
    **if** $\texttt{concat}\Big((v'_{in}, S', v'_{fin}), (v''_{in}, S'', v''_{fin})\Big) = (v_{in}, S, v_{fin})$ **then**
      **return** $\texttt{Check}\exists(\mathcal{K}, \varphi, (v'_{in}, S', v'_{fin}))$
    **else**
      No
  **EndOr**
**else if** $\psi = \langle \mathrm{E} \rangle \varphi$ **then**
  Symmetric to $\psi = \langle \mathrm{B} \rangle \varphi$

a witnessed descriptor element $d = (v_{in}, S, v_{fin})$ and it returns **Yes** if and only if there exists a track $\rho \in \mathrm{Trk}_{\mathcal{K}}$, associated with $d$, such that $\mathcal{K}, \rho \models \psi$. The procedure is recursively defined as follows.

When it is called on a pure propositional formula $\beta$ (base of the recursion), $VAL(\beta, d)$ evaluates $\beta$ over $d$ in the standard way. The evaluation can be performed in deterministic polynomial time, and if $VAL(\beta, d)$ returns $\top$, then there exists a track associated with $d$ (of length at most quadratic in $|W|$) that satisfies $\beta$.

If $\psi = \psi' \vee \psi''$, where $\psi'$ or $\psi''$ feature some temporal modality, the procedure non-deterministically calls itself on $\psi'$ or $\psi''$ (the construct **Either** $c_1$ **Or** $c_2$ **EndOr** denotes a non-deterministic choice between commands $c_1$ and $c_2$).

If $\psi = \langle \mathrm{A} \rangle \psi'$ (respectively, $\langle \overline{\mathrm{A}} \rangle \psi'$), the procedure looks for a new descriptor element for a track starting from the final state (respectively, leading to the initial state) of the current descriptor element $d$. To this aim, we use the procedure $\texttt{aDescrEl}(\mathcal{K}, v, \text{FORW})$ (resp., $\texttt{aDescrEl}(\mathcal{K}, v, \text{BACKW})$) which non-deterministically returns a descriptor element $(v'_{in}, S', v'_{fin})$, with $v'_{in} = v$ (resp., $v'_{fin} = v$), witnessed in $\mathcal{K}$ by exploring forward (resp., backward) the unravelling of $\mathcal{K}$ from $v'_{in}$ (resp., from $v'_{fin}$). Its complexity is polynomial in $|W|$, since it needs to examine the unravelling of $\mathcal{K}$ from $v$ up to depth $2 + |W|^2$.

If $\psi = \langle \mathrm{B} \rangle \psi'$, the procedure looks for a new descriptor element $d_1$ and eventually calls itself on $\psi'$ and $d_1$ only if the current descriptor element $d$ results from the "concatenation" of $d_1$ with a suitable descriptor element $d_2$: if $d_1 = (v'_{in}, S', v'_{fin})$ and $d_2 = (v''_{in}, S'', v''_{fin})$, then $\texttt{concat}(d_1, d_2)$ returns $(v'_{in}, S' \cup \{v'_{fin}, v''_{in}\} \cup S'', v''_{fin})$. Notice that if $\rho_1$ and $\rho_2$ are tracks associated with $d_1$ and $d_2$, respectively, then $\rho_1 \cdot \rho_2$ is associated with $\texttt{concat}(d_1, d_2)$.

The following theorem proves soundness and completeness of the $\texttt{Check}\exists$ procedure.

**Theorem 37.** *For any formula $\psi$ of the fragment $\exists \mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}$ and any witnessed descriptor element $d = (v_{in}, S, v_{fin})$, the procedure $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation if and only if there exists a track $\rho$, associated with $d$, such that $\mathcal{K}, \rho \models \psi$.*

*Proof.* (Soundness) The proof is by induction on the structure of the formula $\psi$.

- $\psi$ is a pure propositional formula $\beta$: let $\rho$ be a witness track for $d$; if $\texttt{Check}\exists(\mathcal{K}, \beta, d)$ has a successful computation, then $VAL(\beta, d)$ is true and so $\mathcal{K}, \rho \models \psi$.

- $\psi = \varphi_1 \vee \varphi_2$: if $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation, then, for some $i \in \{1, 2\}$, $\texttt{Check}\exists(\mathcal{K}, \varphi_i, d)$ has a successful computation. By the inductive hypothesis, there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ associated with $d$ such that $\mathcal{K}, \rho \models \varphi_i$, and thus $\mathcal{K}, \rho \models \varphi_1 \vee \varphi_2$.

- $\psi = \langle \mathrm{A} \rangle \varphi$: if $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation, then there exists a witnessed $d' = (v'_{in}, S', v'_{fin})$, with $v'_{in} = v_{fin}$, such that $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. By the inductive hypothesis, there exists a track $\rho'$, associated with $d'$, such that $\mathcal{K}, \rho' \models \varphi$. If $\rho$ is a track associated with $d$ (which is witnessed by hypothesis), we have that $\mathrm{lst}(\rho) = \mathrm{fst}(\rho') = v_{fin}$ and, by definition, $\mathcal{K}, \rho \models \psi$.

- $\psi = \langle \mathrm{B} \rangle \varphi$: if $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation, then we must distinguish two possible cases.
  ($i$) There exists $d' = (v_{in}, S', v'_{fin})$, witnessed by a track with $(v'_{fin}, v_{fin}) \in \delta$, such that $(v_{in}, S' \cup \{v'_{fin}\}, v_{fin}) = d$, and $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. By the inductive hypothesis, there exists a track $\rho'$, associated with $d'$, such that $\mathcal{K}, \rho' \models \varphi$. Hence

$\mathcal{K}, \rho' \cdot v_{fin} \models \psi$ and $\rho' \cdot v_{fin}$ is associated with $d$.
(*ii*) There exist $d' = (v_{in}, S', v'_{fin})$, witnessed by a track, and $d'' = (v''_{in}, S'', v''_{fin})$, witnessed by a track as well, such that $(v'_{fin}, v''_{in}) \in \delta$, $\texttt{concat}(d', d'') = d$, and $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. By the inductive hypothesis, there exists a track $\rho'$, associated with $d'$, such that $\mathcal{K}, \rho' \models \varphi$. Hence $\mathcal{K}, \rho' \cdot \rho'' \models \psi$, where $\rho''$ is any track associated with $d''$ and $\rho' \cdot \rho''$ is associated with $d$.

The case $\psi = \langle\overline{\mathrm{A}}\rangle\,\varphi$ (respectively, $\psi = \langle\mathrm{E}\rangle\,\varphi$) can be dealt with as $\psi = \langle\mathrm{A}\rangle\,\varphi$ (respectively, $\psi = \langle\mathrm{B}\rangle\,\varphi$).

(Completeness) The proof is by induction on the structure of the formula $\psi$.

- $\psi$ is a pure propositional formula $\beta$: if $\rho$ is associated with $d$ and $\mathcal{K}, \rho \models \beta$, then $VAL(\beta, d) = \top$, and thus $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \varphi_1 \vee \varphi_2$: if there exists a track $\rho$, associated with $d$, such that $\mathcal{K}, \rho \models \varphi_1 \vee \varphi_2$, then $\mathcal{K}, \rho \models \varphi_i$, for some $i \in \{1, 2\}$. By the inductive hypothesis, $\texttt{Check}\exists(\mathcal{K}, \varphi_i, d)$ has a successful computation, and hence $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \langle\mathrm{A}\rangle\,\varphi$: if there exists a track $\rho$, associated with $d$, such that $\mathcal{K}, \rho \models \langle\mathrm{A}\rangle\,\varphi$, then, by definition, there exists a track $\overline{\rho}$, with $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho) = v_{fin}$, such that $\mathcal{K}, \overline{\rho} \models \varphi$. If $d' = (v_{fin}, S', v'_{fin})$ is the descriptor element for $\overline{\rho}$, then, by the inductive hypothesis, $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. Since there exists a computation where the non-deterministic call to $\texttt{aDescrEl}(\mathcal{K}, v_{fin}, \textsc{forw})$ returns the descriptor element $d'$ for $\overline{\rho}$, it follows that $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation.

- $\psi = \langle\mathrm{B}\rangle\,\varphi$: if there exists a track $\rho$, associated with $d$, such that $\mathcal{K}, \rho \models \langle\mathrm{B}\rangle\,\varphi$, there are two possible cases.
(*i*) $\mathcal{K}, \overline{\rho} \models \varphi$, with $\rho = \overline{\rho} \cdot v_{fin}$ for some $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$. If $d' = (v_{in}, S', v'_{fin})$ is the descriptor element for $\overline{\rho}$, by the inductive hypothesis $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. Since there is a computation where $\texttt{aDescrEl}(\mathcal{K}, v_{in}, \textsc{forw})$ returns $d'$ and both $(v'_{fin}, v_{fin}) \in \delta$ and $(v_{in}, S' \cup \{v'_{fin}\}, v_{fin}) = d$, it follows that $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation.
(*ii*) $\mathcal{K}, \overline{\rho} \models \varphi$ with $\rho = \overline{\rho} \cdot \tilde{\rho}$ for some $\overline{\rho}, \tilde{\rho} \in \mathrm{Trk}_{\mathcal{K}}$. Let $d' = (v_{in}, S', v'_{fin})$ and $d'' = (v''_{in}, S'', v''_{fin})$ be the descriptor elements for $\overline{\rho}$ and $\tilde{\rho}$, respectively. Obviously, it holds that $\texttt{concat}(d', d'') = d$. By the inductive hypothesis, $\texttt{Check}\exists(\mathcal{K}, \varphi, d')$ has a successful computation. Since both $\overline{\rho}$ and $\tilde{\rho}$ are witnessed, there is a computation where the calls to $\texttt{aDescrEl}(\mathcal{K}, v_{in}, \textsc{forw})$ and $\texttt{aDescrEl}(\mathcal{K}, v''_{in}, \textsc{forw})$ non-deterministically return $d'$ and $d''$, respectively, and $(v'_{fin}, v''_{in}) \in \delta$ is non-deterministically chosen. Hence, $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ has a successful computation.

The case $\psi = \langle\overline{\mathrm{A}}\rangle\,\varphi$ (respectively, $\psi = \langle\mathrm{E}\rangle\,\varphi$) can be dealt with as $\psi = \langle\mathrm{A}\rangle\,\varphi$ (respectively, $\psi = \langle\mathrm{B}\rangle\,\varphi$). □

It is worth pointing out that $\texttt{Check}\exists(\mathcal{K}, \psi, d)$ cannot deal with $\langle\overline{\mathrm{B}}\rangle$ and $\langle\overline{\mathrm{E}}\rangle$ modalities. To cope with them, descriptor elements are not enough: the whole descriptors must be considered.

We can finally introduce the procedure $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ (Algorithm 5), which searches for counterexamples to the input $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}$ formula $\psi$; indeed, it is possible to prove that it has a successful computation if and only if $\mathcal{K} \not\models \psi$. In the pseudocode of procedure `ProvideCounterex`, $\texttt{to}\exists\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}(\neg\psi)$ denotes the $\exists\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}$ formula equivalent to $\neg\psi$.

**Algorithm 5** $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$

1: $(v_{in}, S, v_{fin}) \leftarrow \texttt{aDescrEl}(\mathcal{K}, w_0, \text{FORW})$ ◁ *$v_{in} = w_0$ is the initial state of $\mathcal{K}$*
2: **return** $\texttt{Check}\exists(\mathcal{K}, \texttt{to}\exists\mathsf{A}\overline{\mathsf{A}}\mathsf{BE}(\neg\psi), (v_{in}, S, v_{fin}))$

On the one hand, if $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ has a successful computation, then there exists a witnessed descriptor element $d = (v_{in}, S, v_{fin})$, where $v_{in}$ is $w_0$ (the initial state of $\mathcal{K}$), such that $\texttt{Check}\exists(\mathcal{K}, \texttt{to}\exists\mathsf{A}\overline{\mathsf{A}}\mathsf{BE}(\neg\psi), d)$ has a successful computation. This means that there exists a track $\rho$, associated with $d$, such that $\mathcal{K}, \rho \models \neg\psi$, and thus $\mathcal{K} \not\models \psi$.

On the other hand, if $\mathcal{K} \not\models \psi$, then there exists an initial track $\rho$ such that $\mathcal{K}, \rho \models \neg\psi$. Let $d$ be the descriptor element for $\rho$: $\texttt{Check}\exists(\mathcal{K}, \texttt{to}\exists\mathsf{A}\overline{\mathsf{A}}\mathsf{BE}(\neg\psi), d)$ has a successful computation. Since $d$ is witnessed by an initial track, some non-deterministic instance of $\texttt{aDescrEl}(\mathcal{K}, w_0, \text{FORW})$ returns $d$. Hence $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ has a successful computation.

As for the complexity, $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ runs in non-deterministic polynomial time (it is in NP), since the number of recursive invocations of the procedure $\texttt{Check}\exists$ is $O(|\psi|)$, and each invocation requires time polynomial in $|W|$ while generating descriptor elements. Therefore, the model checking problem for $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{BE}$ belongs to coNP.

We conclude the section by proving that the model checking problem for $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{BE}$ is coNP-complete. Such a result is an easy corollary of the following theorem.

**Theorem 38.** *Let $\mathcal{K}$ be a finite Kripke structure and $\beta \in \mathsf{Prop}$ be a pure propositional formula. The problem of deciding whether $\mathcal{K} \not\models \beta$ is NP-hard (under LOGSPACE reductions).*

*Proof.* We provide a reduction from the NP-complete SAT problem to the considered problem. Let $\beta$ be a Boolean formula over a set of variables $Var = \{x_1, \ldots, x_n\}$. We build a Kripke structure, $\mathcal{K}_{SAT}^{Var} = (\mathcal{AP}, W, \delta, \mu, w_0)$, with:

- $\mathcal{AP} = Var$;
- $W = \{w_0\} \cup \{w_i^\ell \mid \ell \in \{\top, \bot\},\ 1 \leq i \leq n\}$;
- $\delta = \{(w_0, w_1^\top), (w_0, w_1^\bot)\} \cup \{(w_i^\ell, w_{i+1}^m) \mid \ell, m \in \{\top, \bot\}, 1 \leq i \leq n-1\} \cup \{(w_n^\top, w_n^\top)\} \cup \{(w_n^\bot, w_n^\bot)\}$;
- $\mu(w_0) = \mathcal{AP}$;
- for $1 \leq i \leq n$, $\mu(w_i^\top) = \mathcal{AP}$ and $\mu(w_i^\bot) = \mathcal{AP} \setminus \{x_i\}$.

See Figure 10 for an example of $\mathcal{K}_{SAT}^{Var}$, with $Var = \{x_1, \ldots, x_4\}$.

It is immediate to see that any initial track $\rho$ of any length induces a truth assignment to the variables of $Var$: for any $x_i \in Var$, $x_i$ evaluates to $\top$ if and only if $x_i \in \bigcap_{w \in \text{states}(\rho)} \mu(w)$. Conversely, for any possible truth assignment to the variables in $Var$, there exists an initial track $\rho$ that induces such an assignment: we include in the track the state $w_i^\top$ if $x_i$ is assigned to $\top$, $w_i^\bot$ otherwise.

Let $\gamma = \neg\beta$. It holds that $\beta$ is satisfiable if and only if there exists an initial track $\rho \in \text{Trk}_{\mathcal{K}_{SAT}^{Var}}$ such that $\mathcal{K}_{SAT}^{Var}, \rho \models \beta$, that is, if and only if $\mathcal{K}_{SAT}^{Var} \not\models \gamma$. To conclude, we observe that $\mathcal{K}_{SAT}^{Var}$ can be built with logarithmic working space. □

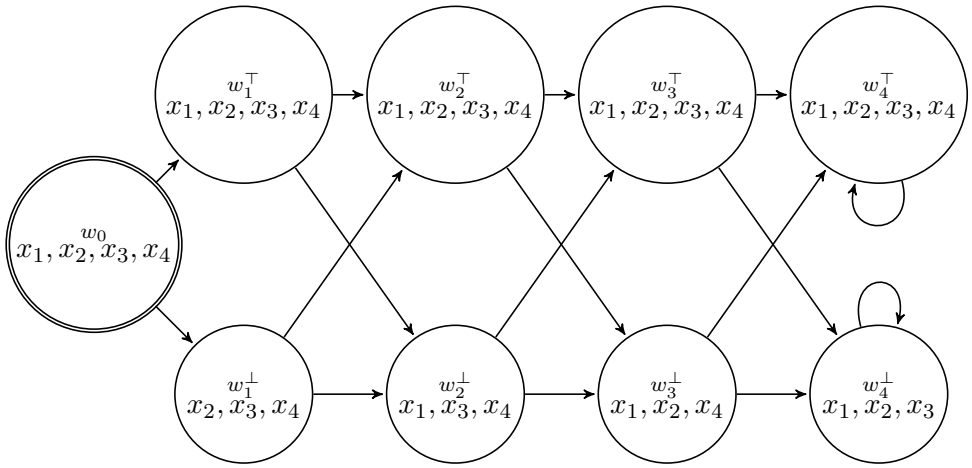



Figure 10: Kripke structure $\mathcal{K}_{SAT}^{Var}$ associated with a SAT formula with variables $Var = \{x_1, x_2, x_3, x_4\}$.

It immediately follows that checking whether $\mathcal{K} \not\models \beta$ for $\beta \in \mathsf{Prop}$ is NP-complete, thus model checking for formulas of $\mathsf{Prop}$ is coNP-complete. Moreover, since a pure propositional formula in $\mathsf{Prop}$ is also a $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{E}}$ formula, $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ is at least as hard as checking whether $\mathcal{K} \not\models \beta$ for $\beta \in \mathsf{Prop}$. Thus, $\texttt{ProvideCounterex}(\mathcal{K}, \psi)$ is NP-complete, hence the model checking problem for $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{E}}$ is coNP-complete.

We conclude the section spending a few words about the complexity of the model checking problem for the fragment $\mathsf{A}\overline{\mathsf{A}}$, also known as *the logic of temporal neighborhood*. As a consequence of the lower bound for $\mathsf{Prop}$, model checking for $\mathsf{A}\overline{\mathsf{A}}$ turns out to be coNP-hard as well. Moreover, the problem is in PSPACE, as $\mathsf{A}\overline{\mathsf{A}}$ is a subfragment of $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{E}}$. Actually, in [23], the authors proved that $\mathsf{A}\overline{\mathsf{A}}$ belongs to $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$ and is $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hard: the complexity class $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ (respectively, $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$) contains the problems decided by a deterministic polynomial time algorithm which requires only $O(\log n)$ (respectively, $O(\log^2 n)$) queries to an NP oracle, being $n$ the input size [12, 32]. Hence, such classes are higher than both NP and coNP in the polynomial time hierarchy.

## 8. Conclusions and future work

In this paper, we have studied the model checking problem for some fragments of Halpern and Shoham's modal logic of time intervals. First, we have considered the large fragment $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{B}}\overline{\mathsf{E}}$, and devised an EXPSPACE model checking algorithm for it, which rests on a contraction method that allows us to restrict the verification of the input formula to a finite subset of tracks of bounded size, called track representatives. We have also proved that the problem is PSPACE-hard, NEXP-hard if a suitable succinct encoding of formulas is allowed. As a matter of fact, in the latter case, the problem can also be proved coNEXP-hard, and thus we conjecture that a tighter lower bound can be established (for instance, EXPSPACE-hardness). Then, we identified some other HS fragments, namely, $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{E}}$, $\forall\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\mathsf{E}$, and $\mathsf{A}\overline{\mathsf{A}}$, whose model checking problem turns out to be (computationally) much simpler than that of full HS and of $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{B}}\overline{\mathsf{E}}$, and comparable to that of point-based temporal logics (as an example, the model checking problem for $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}\overline{\mathsf{E}}$ is PSPACE-complete, and has thus

the same complexity as LTL). Luckily, these fragments are expressive enough to capture meaningful properties of state-transition systems, such as, for instance, mutual exclusion, state reachability, and non-starvation.

One may wonder whether, given the homogeneity assumption, there is the possibility to reduce the model checking problem for HS fragments over finite Kripke structures to a point-based setting. Such an issue has been systematically dealt with in [3]. Together with Laura Bozzelli and Pietro Sala, we consider three semantic variants of HS: the one we introduced in [24] and we used in the subsequent papers, including the present one, called state-based semantics, which allows branching in the past and in the future, the computation-tree-based semantics, allowing branching only in the future, and the linear semantics, disallowing branching. These variants are compared, as for their expressiveness, among themselves and to standard temporal logics, getting a complete picture. In particular, we show that (i) HS with computation-tree-based semantics is equivalent to finitary CTL* and strictly included in HS with state-based semantics, and (ii) HS with linear semantics is equivalent to LTL and incomparable to HS with state-based semantics.

As for future work, we are currently exploring two main research directions. On the one hand, we are looking for other well-behaved fragments of HS; on the other hand, we are thinking of possible ways of relaxing the homogeneity assumption. As for the latter, a promising direction has been recently outlined by Lomuscio and Michaliszyn, who proposed to use regular expressions to define the behavior of proposition letters over intervals in terms of the component states [17]. Our ultimate goal is to be able to deal with interval properties that can only be predicated over time intervals considered as a whole. This is the case, for instance, of temporal aggregations (think of a constraint on the average speed of a moving device during a given time period). In this respect, the existing work on Duration Calculus (DC) model checking seems to be relevant. DC extends interval temporal logic with an explicit notion of state: states are denoted by state expressions and characterized by a duration (the time period during which the system remains in a given state). Recent results on DC model checking and an account of related work can be found in [14].

## Acknowledgements

The work by Adriano Peron has been supported by the SHERPA collaborative project, which has received funding from the European Community 7-th Framework Programme (FP7/2007-2013) under grant agreements ICT-600958. He is solely responsible for its content. The paper does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein. The work by Alberto Molinari and Angelo Montanari has been supported by the GNCS project *Logic, Automata, and Games for Auto-Adaptive Systems*.

# References


[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] H. Bowman and S. J. Thompson. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation*, 13(2):195–239, 2003.

[3] Laura Bozzelli, Alberto Molinari, Angelo Montanari, Adriano Peron, and Pietro Sala. Interval vs. point temporal logic model checking: an expressiveness comparison. In *FSTTCS*, pages 26:1–26:14, 2016.

[4] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. The dark side of interval temporal logic: marking the undecidability border. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):41–83, 2014.

[5] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation*, 20(1):133–166, 2010.

[6] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic*, 161(3):289–304, 2009.

[7] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. What's decidable about Halpern and Shoham's interval logic? The maximal fragment $\mathsf{ABB\overline{B}L}$. In *LICS*, pages 387–396, 2011.

[8] Z. Chaochen and M. R. Hansen. *Duration Calculus - A Formal Approach to Real-Time Systems*. Springer, 2004.

[9] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2002.

[10] F. Giunchiglia and P. Traverso. Planning as model checking. In *ECP*, pages 1–20, 1999.

[11] V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

[12] Georg Gottlob. NP Trees and Carnap's Modal Logic. *Journal of the ACM*, 42(2):421–457, 1995.

[13] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[14] M. R. Hansen, A. D. Phan, and A. W. Brekling. A practical approach to model checking Duration Calculus using Presburger Arithmetic. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):251–278, 2014.

[15] A. R. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *IJCAI*, pages 1010–1016, 2013.

[16] A. R. Lomuscio and J. Michaliszyn. Decidability of model checking multi-agent systems against a class of EHS specifications. In *ECAI*, pages 543–548, 2014.

[17] A. R. Lomuscio and J. Michaliszyn. Model checking multi-agent systems against epistemic HS specifications with regular expressions. In *KR*, pages 298–308, 2016.

[18] A. R. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *TACAS*, pages 450–454, 2006.

[19] Jerzy Marcinkowski and Jakub Michaliszyn. The undecidability of the logic of subintervals. *Fundamenta Informaticae*, 131(2):217–240, 2014.

[20] A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. Checking interval properties of computations. *Acta Informatica*, 53(6-8):587–619, 2016.

[21] A. Molinari, A. Montanari, and A. Peron. Complexity of ITL model checking: some well-behaved fragments of the interval logic HS. In *TIME*, pages 90–100, 2015.

[22] A. Molinari, A. Montanari, and A. Peron. A model checking procedure for interval temporal logics based on track representatives. In *CSL*, pages 193–210, 2015.

[23] A. Molinari, A. Montanari, A. Peron, and P. Sala. Model Checking Well-Behaved Fragments of HS: the (Almost) Final Picture. In *KR*, pages 473–483, 2016.

[24] A. Montanari, A. Murano, G. Perelli, and A Peron. Checking interval properties of computations. In *TIME*, pages 59–68, 2014.

[25] A. Montanari, G. Puppis, and P. Sala. Maximal decidable fragments of Halpern and Shoham's modal logic of intervals. In *ICALP*, pages 345–356, 2010.

[26] Angelo Montanari and Pietro Sala. Interval-based synthesis. In *GandALF*, pages 102–115, 2014.

[27] B. Moszkowski. *Reasoning About Digital Circuits*. PhD thesis, Department of Computer Science, Stanford University, Stanford, CA, 1983.

[28] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[29] R. J. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

[30] I. Pratt-Hartmann. Temporal prepositions and their logic. *Artificial Intelligence*, 166(1-2):1–36, 2005.

[31] P. Roeper. Intervals and tenses. *Journal of Philosophical Logic*, 9:451–469, 1980.

[32] Ph. Schnoebelen. Oracle circuits for branching-time model checking. In *ICALP*, pages 790–801, 2003.

[33] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 2012.

[34] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.

[35] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

# Appendix A. Proofs

## Appendix A.1. Proof of Lemma 12

In the proof, we will exploit the fact that if two tracks in $\mathrm{Trk}_{\mathcal{K}}$ have the same $B_{k+1}$-descriptor, then they also have the same $B_k$-descriptor. The latter can indeed be obtained from the former by removing the nodes at depth $k+1$ (leaves) and then deleting isomorphic subtrees possibly originated by the removal.

*Proof.* By induction on $k \geq 0$.

*Base case* ($k = 0$): let us assume $\rho_1$ and $\rho_2$ are associated with the descriptor element $(v_{in}, S, v_{fin})$ and $\rho'_1$ and $\rho'_2$ with $(v'_{in}, S', v'_{fin})$. Thus $\rho_1 \cdot \rho'_1$ and $\rho_2 \cdot \rho'_2$ are both described by the descriptor element $(v_{in}, S \cup \{v_{fin}, v'_{in}\} \cup S', v'_{fin})$.

*Inductive step* ($k > 0$): let $\mathcal{D}_{B_k}$ be the $B_k$-descriptor for $\rho_1 \cdot \rho'_1$ and $\mathcal{D}'_{B_k}$ be the one for $\rho_2 \cdot \rho'_2$: their roots are the same, as for $k = 0$; let us now consider a prefix $\rho$ of $\rho_1 \cdot \rho'_1$:

- if $\rho$ is a proper prefix of $\rho_1$, since $\rho_1$ and $\rho_2$ have the same $B_k$-descriptor, there exists a prefix $\overline{\rho}$ of $\rho_2$ associated with the same subtree as $\rho$ of depth $k-1$ in the descriptor for $\rho_1$ (and $\rho_2$);
- for $\rho = \rho_1$, it holds that $\rho_1$ and $\rho_2$ have the same $B_{k-1}$-descriptor because they have the same $B_k$-descriptor;
- if $\rho$ is a proper prefix of $\rho_1 \cdot \rho'_1$ such that $\rho = \rho_1 \cdot \tilde{\rho}_1$ for some prefix $\tilde{\rho}_1$ of $\rho'_1$, then two cases have to be taken into account:
  - if $|\tilde{\rho}_1| = 1$, then $\tilde{\rho}_1 = v'_{in}$; but also $\mathrm{fst}(\rho'_2) = v'_{in}$. Let us now consider the $B_{k-1}$-descriptors for $\rho_1 \cdot v'_{in}$ and $\rho_2 \cdot v'_{in}$: the labels of the roots are the same, namely $(v_{in}, S \cup \{v_{fin}\}, v'_{in})$, then the subtrees of depth $k-2$ are exactly the same as in $\rho_1$ and $\rho_2$'s $B_{k-1}$-descriptor, (possibly) with the addition of the $B_{k-2}$-descriptor for $\rho_1$ (which is equal to that for $\rho_2$). Thus $\rho_1 \cdot v'_{in}$ and $\rho_2 \cdot v'_{in}$ have the same $B_{k-1}$-descriptor;
  - otherwise, since $\tilde{\rho}_1$ is a prefix of $\rho'_1$ of length at least 2, and $\rho'_1$ and $\rho'_2$ have the same $B_k$-descriptor, there exists a prefix $\tilde{\rho}_2$ of $\rho'_2$ associated with the same subtree of depth $k-1$ as $\tilde{\rho}_1$ (in the $B_k$-descriptor for $\rho'_1$). Hence, by inductive hypothesis, $\rho_1 \cdot \tilde{\rho}_1$ and $\rho_2 \cdot \tilde{\rho}_2$ have the same $B_{k-1}$-descriptor.

Therefore we have shown that for any proper prefix of $\rho_1 \cdot \rho'_1$ there exists a proper prefix of $\rho_2 \cdot \rho'_2$ having the same $B_{k-1}$-descriptor. The inverse can be shown by symmetry. Thus $\mathcal{D}_{B_k}$ is equal to $\mathcal{D}'_{B_k}$. □

## Appendix A.2. Proof of Theorem 29

*Proof.* The proof is by induction on $i \geq u+1$.
(Case $i = u+1$) We consider two cases:

1. if $\rho_{ds}(u) = \rho_{ds}(u+1) = d \in \mathcal{C}$, then we have $Q_{-2}(u) = \mathcal{C} \setminus \{d\}$, and $Q_{-1}(u) = \{d\}$, $Q_0(u) = Q_1(u) = \cdots = Q_s(u) = \emptyset$. Moreover, it holds that $Q_{-2}(u+1) = \mathcal{C} \setminus \{d\}$, $Q_{-1}(u) = \emptyset$, $Q_0(u) = \{d\}$, and $Q_1(u) = Q_2(u) = \cdots = Q_s(u) = \emptyset$. $c(u) >_{lex} c(u+1)$ and the thesis follows.
2. if $d, d' \in \mathcal{C}$, with $d \neq d'$, $\rho_{ds}(u) = d$, and $\rho_{ds}(u+1) = d'$, then we have $Q_{-2}(u) = \mathcal{C} \setminus \{d\}$, $Q_{-1}(u) = \{d\}$, and $Q_0(u) = Q_1(u) = \cdots = Q_s(u) = \emptyset$. Moreover, it holds that $Q_{-2}(u+1) = \mathcal{C} \setminus \{d, d'\}$, $Q_{-1}(u) = \{d, d'\}$, $Q_0(u) = Q_1(u) = \cdots = Q_s(u) = \emptyset$, and $c(u) >_{lex} c(u+1)$, implying the thesis.

(Case $i > u+1$) In the following, we say that $\rho_{ds}(\ell)$ and $\rho_{ds}(m)$ ($\ell < m$) are consecutive occurrences of a descriptor element $d$ if there are no other occurrences of $d$ in $\rho_{ds}(\ell+1, m-1)$. We consider the following cases:

1. If $\rho_{ds}(i)$ is the first occurrence of $d \in \mathcal{C}$, then $d \in Q_{-2}(i-1)$, $d \in Q_{-1}(i)$, and it holds that $c(i-1) >_{lex} c(i)$.
2. If $\rho_{ds}(i)$ is the second occurrence of $d \in \mathcal{C}$, according to the definition, $\rho_{ds}(i)$ can not be 1-indistinguishable from the previous occurrence of $d$, and thus $d \in Q_{-1}(i-1)$ ($\rho_{ds}(u, i-1)$ contains the first occurrence of $d$) and $d \in Q_0(i)$, proving that $c(i-1) >_{lex} c(i)$.
3. If $\rho_{ds}(i)$ is at least the third occurrence of $d \in \mathcal{C}$, but $\rho_{ds}(i)$ is *not* 1-indistinguishable from the immediately preceding occurrence of $d$, $\rho_{ds}(i')$, with $i' < i$, then $DElm(\rho_{ds}(u, i'-1)) \subset DElm(\rho_{ds}(u, i-1))$. Hence, there exists a first occurrence of some $d' \in \mathcal{C}$ in $\rho_{ds}(i'+1, i-1)$, say $\rho_{ds}(j) = d'$, for $i'+1 \leq j \leq i-1$. Thus, $d \in Q_{-1}(j), \cdots, d \in Q_{-1}(i-1)$, and $d \in Q_0(i)$, proving that $c(i-1) >_{lex} c(i)$.
4. In the remaining cases, we assume that $\rho_{ds}(i)$ is *at least the third occurrence* of $d \in \mathcal{C}$. If $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are both occurrences of $d \in \mathcal{C}$ and $\rho_{ds}(i-1)$ is $t$-indistinguishable, for some $t > 0$, and not $(t+1)$-indistinguishable, from the immediately preceding occurrence of $d$, then $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are exactly $(t+1)$-indistinguishable. Thus, $d \in Q_t(i-1)$ and $d \in Q_{t+1}(i)$, implying that $c(i-1) >_{lex} c(i)$ (as a particular case, if $\rho_{ds}(i-1)$ and the immediately preceding occurrence are not 1-indistinguishable, then $\rho_{ds}(i-1)$ and $\rho_{ds}(i)$ are at most 1-indistinguishable).
5. If $\rho_{ds}(i)$ is exactly 1-indistinguishable from the immediately preceding occurrence of $d$, $\rho_{ds}(j)$, with $j < i-1$, then $DElm(\rho_{ds}(u, j-1)) = DElm(\rho_{ds}(u, i-1))$, and there are no first occurrences of any $d' \in \mathcal{C}$ in $\rho_{ds}(j, i-1)$. If $\rho_{ds}(j)$ is not 1-indistinguishable from its previous occurrence of $d$, it immediately follows that $d \in Q_0(j), \cdots, d \in Q_0(i-1)$ and $d \in Q_1(i)$, implying that $c(i-1) >_{lex} c(i)$.
   Otherwise, there exists $j < i' < i$ such that $\rho_{ds}(i') = d'' \in \mathcal{C}$ is not 1-indistinguishable from any occurrence of $d''$ before $j$ (as a matter of fact, if this was not the case, $\rho_{ds}(i)$ and $\rho_{ds}(j)$ would be 2-indistinguishable); in particular, $\rho_{ds}(i')$ is not 1-indistinguishable from the last occurrence of $d''$ before $j$, say $\rho_{ds}(j')$, for some $j' < j$ (such a $j'$ exists since there are no first occurrences in $\rho_{ds}(j+1, i-1)$). Now, if by contradiction every pair of consecutive occurrences of $d''$ in $\rho_{ds}(j', i')$ were 1-indistinguishable, then by Corollary 26 $\rho_{ds}(j')$ and $\rho_{ds}(i')$ would be 1-indistinguishable. Thus, a pair of consecutive occurrences of $d''$ exists, where the second element in the pair is $\rho_{ds}(\ell) = d''$, with $j < \ell < i$, such that they are not 1-indistinguishable. By inductive hypothesis, $d'' \in Q_{-1}(\ell-1)$ and $d'' \in Q_0(\ell)$. Therefore, $d \in Q_0(\ell), \cdots, d \in Q_0(i-1)$ (recall that there are no first occurrences between $j$ and $i$) and $d \in Q_1(i)$, proving that $c(i-1) >_{lex} c(i)$.
6. If $\rho_{ds}(j) = d \in \mathcal{C}$ is at most $t$-indistinguishable (for some $t \geq 1$) from a preceding occurrence of $d$ and $\rho_{ds}(j)$ and $\rho_{ds}(i) = d$, with $j < i-1$, are $(t+1)$-indistinguishable consecutive occurrences of $d$ (by definition of indistinguishability, $\rho_{ds}(j)$ and $\rho_{ds}(i)$ can not be more than $(t+1)$-indistinguishable), any occurrence of $d' \in \mathcal{C}$ in $\rho_{ds}(j+1, i-1)$ is (at least) $t$-indistinguishable from another occurrence of $d'$ before $j$. By Proposition 24, all pairs of consecutive occurrences of $d'$ in $\rho_{ds}(j+1, i-1)$ are (at least) $t$-indistinguishable, hence $d \in Q_t(j), \cdots, d \in Q_t(i-1)$ and finally $d \in Q_{t+1}(i)$, proving that $c(i-1) >_{lex} c(i)$.
7. If $\rho_{ds}(j) = d \in \mathcal{C}$ is at most $t$-indistinguishable (for some $t \geq 1$) from a preceding occurrence of $d$, and $\rho_{ds}(j)$ and $\rho_{ds}(i) = d$, with $j < i-1$, are consecutive occurrences of $d$ which are at most

$\overline{t}$-indistinguishable, for some $1 \leq \overline{t} \leq t$, we preliminarily observe that $DElm(\rho_{ds}(u, j-1)) = DElm(\rho_{ds}(u, i-1))$. Then, if some $d'' \in \mathcal{C}$, with $d'' \neq d$, occurs in $\rho_{ds}(j+1, i-1)$ and it is not 1-indistinguishable from any occurrence of $d''$ before $j$, then $\overline{t} = 1$ and we are again in case 5.

Otherwise, all the occurrences of descriptor elements in $\rho_{ds}(j+1, i-1)$ are (at least) 1-indistinguishable from other occurrences before $j$. Moreover, there exists $j < i' < i$ such that $\rho_{ds}(i') = d' \in \mathcal{C}, d \neq d'$, and it is at most $(\overline{t}-1)$-indistinguishable from another occurrence of $d'$ before $j$. Analogously to case 5, by Proposition 24, $\rho_{ds}(i')$ must be $(\overline{t}-1)$-indistinguishable from the last occurrence of $d'$ before $j$, say $\rho_{ds}(j')$, with $j' < j$. But two consecutive occurrences of $d'$ in $\rho_{ds}(j', i')$ must then be at most $(\overline{t}-1)$-indistinguishable (if all pairs of occurrences of $d'$ in $\rho_{ds}(j', i')$ were $\overline{t}$-indistinguishable, $\rho_{ds}(i')$ and $\rho_{ds}(j')$ would be $\overline{t}$-indistinguishable as well), where the second occurrence is $\rho_{ds}(\ell) = d'$ for some $j < \ell \leq i'$. By applying the inductive hypothesis, we have $d' \in Q_{\overline{t}-2}(\ell - 1)$ and $d' \in Q_{\overline{t}-1}(\ell)$. As a consequence, we have $d \in Q_{\overline{t}-1}(\ell)$, $\cdots$, $d \in Q_{\overline{t}-1}(i-1)$ (all descriptor elements in $\rho_{ds}(j, i)$ are at least $(\overline{t}-1)$-indistinguishable from other occurrences before $j$) and finally $d \in Q_{\overline{t}}(i)$, implying that $c(i-1) >_{lex} c(i)$. □

It is worth pointing out that, from the proof of the theorem, it follows that the definition of $f$ is in fact redundant: cases (c) and (e) never occur.

## Appendix A.3. Proof of Lemma 33

*Proof.* The proof is by induction on the structure of $\psi$. The cases in which $\psi = \top$, $\psi = \bot$, $\psi = p \in \mathcal{AP}$ are trivial. The cases in which $\psi = \neg\varphi$, $\psi = \varphi_1 \wedge \varphi_2$ are also trivial and omitted. We focus on the remaining cases.

- $\psi = \langle \mathrm{A} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, then there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\tilde{\rho}) = \mathrm{fst}(\rho)$ and $\mathcal{K}, \rho \models \varphi$. By Theorem 32 the unravelling procedure returns $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\overline{\rho}) = \mathrm{fst}(\rho)$ and $\overline{\rho}$ and $\rho$ have the same $B_k$-descriptor, thus $\mathcal{K}, \overline{\rho} \models \varphi$. By the inductive hypothesis, $\mathtt{Check}(\mathcal{K}, k, \varphi, \overline{\rho}) = 1$, hence $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$.

  Vice versa, if $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{lst}(\tilde{\rho}) = \mathrm{fst}(\rho)$ and $\mathtt{Check}(\mathcal{K}, k, \varphi, \rho) = 1$. By the inductive hypothesis, $\mathcal{K}, \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.

- $\psi = \langle \overline{\mathrm{A}} \rangle \varphi$. The proof is symmetric to the case $\psi = \langle \mathrm{A} \rangle \varphi$.

- $\psi = \langle \mathrm{B} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, there exists $\rho \in \mathrm{Pref}(\tilde{\rho})$ such that $\mathcal{K}, \rho \models \varphi$. By the inductive hypothesis, $\mathtt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. Since all prefixes of $\tilde{\rho}$ are checked, $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. *Note that, by definition of descriptor, if $\tilde{\rho}$ is a track representative of a $B_k$-descriptor $\mathcal{D}_{B_k}$, a prefix of $\tilde{\rho}$ is a representative of a $B_{k-1}$-descriptor, whose root is a child of the root of $\mathcal{D}_{B_k}$.*

  Vice versa, if $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, then for some track $\rho \in \mathrm{Pref}(\tilde{\rho})$, we have $\mathtt{Check}(\mathcal{K}, k-1, \varphi, \rho) = 1$. By the inductive hypothesis $\mathcal{K}, \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.

- $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$. If $\mathcal{K}, \tilde{\rho} \models \psi$, then there exists $\rho$ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ for which $\mathcal{K}, \tilde{\rho} \cdot \rho \models \varphi$. If $|\rho| = 1$, since by the inductive hypothesis $\mathtt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$, then $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. Otherwise, the unravelling algorithm returns a track $\overline{\rho}$ with the same $B_k$-descriptor as $\rho$. Thus, by the extension Proposition 13, $\tilde{\rho} \cdot \rho$ and $\tilde{\rho} \cdot \overline{\rho}$ have the same $B_k$-descriptor. Thus $\mathcal{K}, \tilde{\rho} \cdot \overline{\rho} \models \varphi$. So (by inductive hypothesis) $\mathtt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \overline{\rho}) = 1$ implying that $\mathtt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$. *Note that, given two tracks $\rho, \rho'$ of $\mathcal{K}$, if we are considering $\overline{\rho}$ as the*

*track representative of the $B_k$-descriptor of $\rho$, and the unravelling algorithm returns $\overline{\rho}'$ as the representative of the $B_k$-descriptor of $\rho'$, since by Lemma 12 $\rho \cdot \rho'$ and $\overline{\rho} \cdot \overline{\rho}'$ have the same $B_k$-descriptor, we have that $\overline{\rho} \cdot \overline{\rho}'$ is the representative of the $B_k$-descriptor of $\rho \cdot \rho'$.*

Vice versa, if $\texttt{Check}(\mathcal{K}, k, \psi, \tilde{\rho}) = 1$, there exists $\rho$ such that $\tilde{\rho} \cdot \rho \in \mathrm{Trk}_{\mathcal{K}}$ and $\texttt{Check}(\mathcal{K}, k, \varphi, \tilde{\rho} \cdot \rho) = 1$. By the inductive hypothesis, $\mathcal{K}, \tilde{\rho} \cdot \rho \models \varphi$, hence $\mathcal{K}, \tilde{\rho} \models \psi$.

- $\psi = \langle \overline{\mathrm{E}} \rangle \varphi$. The proof is symmetric to the case $\psi = \langle \overline{\mathrm{B}} \rangle \varphi$. □

## Appendix A.4. Proof of Theorem 34

*Proof.* If $\mathcal{K} \models \psi$, then for all $\rho \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\rho) = w_0$ is the initial state of $\mathcal{K}$, we have $\mathcal{K}, \rho \models \psi$. By Lemma 33, it follows that $\texttt{Check}(\mathcal{K}, \mathrm{Nest}_{\mathrm{B}}(\psi), \psi, \rho) = 1$. Now, the unravelling procedure returns a subset of the initial tracks. This implies that $\texttt{ModCheck}(\mathcal{K}, \psi) = 1$.

On the other hand, if $\texttt{ModCheck}(\mathcal{K}, \psi) = 1$, then for any track $\rho$ with $\mathrm{fst}(\rho) = w_0$ *returned by the unravelling algorithm*, $\texttt{Check}(\mathcal{K}, \mathrm{Nest}_{\mathrm{B}}(\psi), \psi, \rho) = 1$ and, by Lemma 33, $\mathcal{K}, \rho \models \psi$. Assume now that a track $\tilde{\rho}$, with $\mathrm{fst}(\tilde{\rho}) = w_0$, is *not* returned by the unravelling algorithm. By Theorem 32, there exists a track $\overline{\rho}$, with $\mathrm{fst}(\overline{\rho}) = w_0$, which is returned in place of $\tilde{\rho}$ and $\overline{\rho}$ has the same $B_k$-descriptor as $\tilde{\rho}$ (with $k = \mathrm{Nest}_{\mathrm{B}}(\psi)$). Since $\mathcal{K}, \tilde{\rho} \models \psi \iff \mathcal{K}, \overline{\rho} \models \psi$ (by Theorem 14) and $\mathcal{K}, \overline{\rho} \models \psi$, we get that $\mathcal{K}, \tilde{\rho} \models \psi$. So all tracks starting from state $w_0$ model $\psi$, implying that $\mathcal{K} \models \psi$. □

## Appendix A.5. NEXP-hardness of succinct $\mathsf{A\overline{A}B\overline{B}E}$

In Section 5, we proved that the model checking problem for $\mathsf{A\overline{A}B\overline{B}E}$ formulas is in EXPSPACE, and, in Section 6, that it is PSPACE-hard. Here we prove that the model checking problem for $\mathsf{A\overline{A}B\overline{B}E}$ is in between EXPSPACE and NEXP when a suitable encoding of formulas is exploited. Such an encoding is *succinct*, in the sense that the following binary-encoded shorthands are used: $\langle \mathrm{B} \rangle^k \psi$ stands for $k$ repetitions of $\langle \mathrm{B} \rangle$ before $\psi$, where $k$ is represented in binary (the same for all the other HS modalities); moreover, $\bigwedge_{i=l,\cdots,r} \psi(i)$ denotes a conjunction of formulas which contain some occurrences of the index $i$ as exponents ($l$ and $r$ are binary encoded naturals), e.g., $\bigwedge_{i=1,\cdots,5} \langle \mathrm{B} \rangle^i \top$. Finally, we denote by $\mathrm{expand}(\psi)$ the expanded form of $\psi$, where all exponents $k$ are removed from $\psi$, by explicitly repeating $k$ times each HS modality with such an exponent, and big conjunctions are replaced by conjunctions of formulas without indexes.

It is not difficult to show that there exists a constant $c > 0$ such that, for all succinct $\mathsf{A\overline{A}B\overline{B}E}$ formulas $\psi$, $|\mathrm{expand}(\psi)| \leq 2^{|\psi|^c}$. Therefore the model checking algorithm $\texttt{ModCheck}$ of Section 5 still runs in *exponential working space* with respect to the succinct input formula $\psi$—by preliminarily expanding $\psi$ to $\mathrm{expand}(\psi)$—as $\tau(|W|, \mathrm{Nest}_{\mathrm{B}}(\mathrm{expand}(\psi)))$ is exponential in $|W|$ and $|\psi|$.

Moreover, the following result holds:

**Theorem 39.** *The model checking problem for succinctly encoded formulas of $\mathsf{A\overline{A}B\overline{B}E}$ over finite Kripke structures is NEXP-hard (under polynomial-time reductions).*

The theorem is proved by means of a reduction from the acceptance problem for a (generic) language $L$ decided by a *non-deterministic one-tape* Turing machine $M$ (w.l.o.g.) that halts in $O(2^{n^k})$ computation steps on any input of size $n$, where $k > 0$ is a constant. We suitably define a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and a succinct $\mathsf{A\overline{A}B\overline{B}E}$ formula $\psi$ such that $\mathcal{K} \models \psi$ if and only if $M$ accepts its input string $c_0 c_1 \cdots c_{n-1}$.

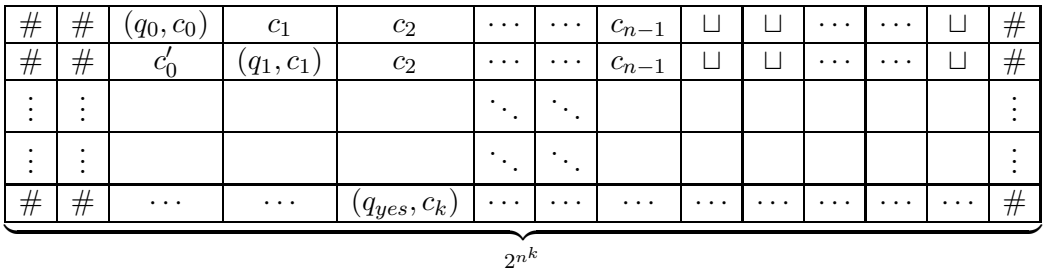



Figure A.11: An example of computation table (tableau).

This allows us to conclude that the model checking problem for succinct $\mathsf{A\overline{A}B\overline{B}E}$ formulas over finite Kripke structures is between NEXP and EXPSPACE. We end this section by proving Theorem 39.

*Proof.* Let us consider a language $L$ decided by a *non-deterministic one-tape* Turing machine $M$ (w.l.o.g.) that halts after no more than $2^{n^k} - 3$ computation steps on an input of size $n$ (assuming a sufficiently high constant $k \in \mathbb{N}$). Hence, $L$ belongs to NEXP.

Let $\Sigma$ and $Q$ be the alphabet and the set of states of $M$, respectively, and let $\#$ be a special symbol not in $\Sigma$ used as separator for configurations (in the following we let $\Sigma' = \Sigma \cup \{\#\}$). The alphabet $\Sigma$ is assumed to contain the blank symbol $\sqcup$.

As usual, a computation of $M$ is a sequence of configurations of $M$, where each configuration fixes the content of the tape, the position of the head on the tape and the internal state of $M$. We use a standard encoding for computations called *computation table* (or tableau) (see [28, 33] for further details). Each configuration of $M$ is a sequence over the alphabet $\Gamma = \Sigma' \cup (Q \times \Sigma)$; a symbol in $(q, c) \in Q \times \Sigma$ occurring in the $i$-th position encodes the fact that the machine has internal state $q$ and its head is currently on the $i$-th position of the tape (obviously exactly one occurrence of a symbol in $Q \times \Sigma$ occurs in each configuration). Since $M$ halts after no more than $2^{n^k} - 3$ computation steps, $M$ uses at most $2^{n^k} - 3$ cells on its tape, so the size of a configuration is $2^{n^k}$ (we need 3 occurrences of the auxiliary symbol $\#$, two for delimiting the beginning of the configuration, and one for the end; additionally $M$ never overwrites delimiters $\#$). If a configuration is actually shorter than $2^{n^k}$, it is padded with $\sqcup$ symbols in order to reach length $2^{n^k}$ (which is a fixed number, once the input length is known). Moreover, since $M$ halts after no more than $2^{n^k} - 3$ computation steps, the number of configurations is $2^{n^k} - 3$. The computation table is basically a matrix of $2^{n^k} - 3$ rows and $2^{n^k}$ columns, where the $i$-th row records the configuration of $M$ at the $i$-th computation step.

As an example, a possible table is depicted in Figure A.11. In the first configuration (row) the head is in the leftmost position (on the right of delimiters $\#$) and $M$ is in state $q_0$. In addition, we have the string symbols $c_0 c_1 \cdots c_{n-1}$ padded with occurrences of $\sqcup$ to reach length $2^{n^k}$. In the second configuration, the head has moved one position to the right, $c_0$ has been overwritten by $c'_0$, and $M$ is in state $q_1$. From the first two rows, we can deduce that the tuple $(q_0, c_0, q_1, c'_0, \rightarrow)$ belongs to the transition relation $\delta_M$ of $M$ (we assume that $\delta_M \subseteq Q \times \Sigma \times Q \times \Sigma \times \{\rightarrow, \leftarrow, \bullet\}$ with the obvious standard meaning).

Following [28, 33], we now introduce the notion of (legal) window. A window is a $2 \times 3$ matrix, in which the first row represents three consecutive symbols of a possible configuration.

The second row represents the three symbols which are placed exactly in the same position in the next configuration. A window is legal when the changes from the first to the second row are coherent with $\delta_M$ in the obvious sense. Actually, the set of legal windows, which we denote by $Wnd \subseteq \left(\Gamma^3\right)^2$, is a tabular representation of the transition relation $\delta_M$.

For example, two legal windows associated with the table of the previous example are:

| | | |
|---|---|---|
| $\#$ | $(q_0, c_0)$ | $c_1$ |
| $\#$ | $c_0'$ | $(q_1, c_1)$ |

| | | |
|---|---|---|
| $(q_0, c_0)$ | $c_1$ | $c_2$ |
| $c_0'$ | $(q_1, c_1)$ | $c_2$ |

Formally, a $((x, y, z), (x', y', z')) \in Wnd$ can be represented as

| | | |
|---|---|---|
| $x$ | $y$ | $z$ |
| $x'$ | $y'$ | $z'$ |

with $x, x', y, y', z, z' \in \Gamma$,

where the following constraints must hold:

1. if all $x, y, z \in \Sigma'$ ($x$, $y$, $z$ are not state-symbol pairs), then $y = y'$;
2. if one of $x$, $y$ and $z$ belongs to $Q \times \Sigma$, then $x'$, $y'$ and $z'$ are coherent with $\delta_M$, and
3. $(x = \# \Rightarrow x' = \#) \wedge (y = \# \Rightarrow y' = \#) \wedge (z = \# \Rightarrow z' = \#)$.

As we said, $M$ never overwrites a $\#$ and we can assume that the head never visits a $\#$, as well (some more windows can be possibly added if necessary, see [28]).

In the following we define a Kripke structure $\mathcal{K} = (\mathcal{AP}, W, \delta, \mu, w_0)$ and a (succinct) formula $\psi$ of $\mathsf{A\overline{A}B\overline{B}E}$ such that $\mathcal{K} \models \psi$ if and only if $M$ accepts its input string $c_0 c_1 \cdots c_{n-1}$. The set of propositional letters is $\mathcal{AP} = \Gamma \cup \Gamma^3 \cup \{start\}$. The Kripke structure $\mathcal{K}$ is obtained by suitably composing a basic pattern called *gadget*. An instance of the gadget is associated with a triple of symbols $(a, b, c) \in \Gamma^3$ (i.e., a sequence of three adjacent symbols in a configuration) and consists of 3 states: $q^0_{(a,b,c)}$, $q^1_{(a,b,c)}$, $q^2_{(a,b,c)}$ such that

$$\mu\left(q^0_{(a,b,c)}\right) = \mu\left(q^1_{(a,b,c)}\right) = \{(a, b, c), c\} \text{ and } \mu\left(q^2_{(a,b,c)}\right) = \emptyset.$$

Moreover,

$$\delta\left(q^0_{(a,b,c)}\right) = \left\{q^1_{(a,b,c)}\right\} \text{ and } \delta\left(q^1_{(a,b,c)}\right) = \left\{q^2_{(a,b,c)}\right\}.$$

(See Figure A.12.) The underlying idea is that a gadget associated with $(x, y, z) \in \Gamma^3$ "records" the current proposition letter $z$, as well as two more "past" letters ($x$ and $y$).

The Kripke structure $\mathcal{K}$ has (an instance of) a gadget for every $(x, y, z) \in \Gamma^3$ and for all $(x, y, z)$ and $(x', y', z')$ in $\Gamma^3$, we have $q^0_{(x',y',z')} \in \delta\left(q^2_{(x,y,z)}\right)$ if and only if $x' = y$ and $y' = z$. Moreover, $\mathcal{K}$ has some additional (auxiliary) states $w_0, \cdots, w_6$ described in Figure A.13 and $\delta(w_6) = \left\{q^0_{(\#,\#,x)} \mid x \in \Gamma\right\}$. Note that the overall size of $\mathcal{K}$ only depends on $|\Gamma|$ and it is constant w.r. to the input string $c_0 c_1 \cdots c_{n-1}$ of $M$.

Now we want to decide whether an input string belongs to the language $L$ by solving the model checking problem $\mathcal{K} \models start \rightarrow \langle \mathrm{A} \rangle \xi$ where $\xi$ is satisfied only by tracks which represent a successful computation of $M$. Since the only (initial) track which satisfies $start$ is $w_0 w_1$, we are actually verifying the existence of a track which begins with $w_1$ and satisfies $\xi$.

As for $\xi$, it requires that a track $\rho$, for which $\mathcal{K}, \rho \models \xi$ (with $\mathrm{fst}(\rho) = w_1$), mimics a successful computation of $M$ in this way: every interval $\rho(i, i+1)$, for $i \mod 3 = 0$, satisfies the proposition

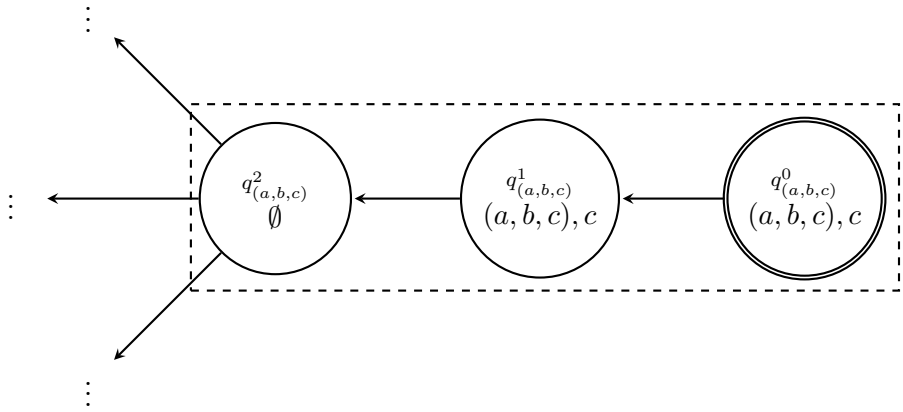



Figure A.12: An instance of the described gadget for $(a, b, c) \in \Gamma^3$.

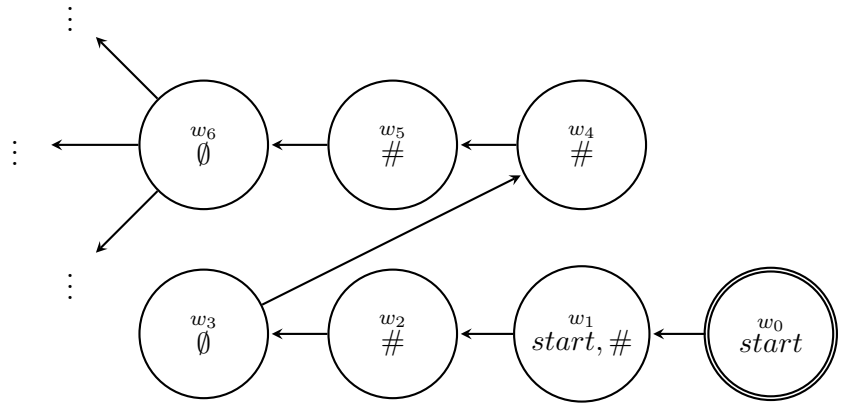



Figure A.13: Initial part of $\mathcal{K}$.

letter $p \in \mathcal{AP}$ if and only if the $\frac{i}{3}$-th character of the computation represented by $\rho$ is $p$ (note that as a consequence of the gadget structure, only $\rho$'s subtracks $\overline{\rho} = \rho(i, i+1)$ for $i \mod 3 = 0$ can satisfy some proposition letters). A symbol of a configuration is mapped to an occurrence of an instance of a gadget in $\rho$; $\rho$, in turn, encodes a computation of $M$ through the concatenation of the first, second, third... rows of the computation table (two consecutive configurations are separated by 3 occurrences of #, which require 9 states overall).

Let us now define the HS formula $\xi = \psi_{accept} \wedge \psi_{input} \wedge \psi_{window}$, where

$$\psi_{accept} = \langle \mathrm{B} \rangle \langle \mathrm{A} \rangle \bigvee_{a \in \Sigma} (q_{yes}, a)$$

requires a track to contain an occurrence of the accepting state of $M$, $q_{yes}$; $\psi_{input}$ is a bit more involved and demands that the subtrack corresponding to the first configuration of $M$ actually "spells" the input $c_0 c_1 \cdots c_{n-1}$, suitably padded with occurrences of $\sqcup$ and terminated by a # (in the following, $\ell(k)$, introduced in Example 1, is satisfied only by those tracks whose length equals

$k$ ($k \geq 2$) and it has a binary encoding of $O(\log k)$ bits):

$$\psi_{input} = [B]\Big(\ell(7) \to \langle \mathrm{A}\rangle(q_0, c_0)\Big) \wedge [B]\Big(\ell(10) \to \langle \mathrm{A}\rangle\, c_1\Big) \wedge [B]\Big(\ell(13) \to \langle \mathrm{A}\rangle\, c_2\Big) \wedge$$
$$\vdots$$
$$[B]\Big(\ell(7 + 3(n-1)) \to \langle \mathrm{A}\rangle\, c_{n-1}\Big) \wedge$$
$$[B]\Bigg(\langle \mathrm{B}\rangle^{5+3n} \top \wedge [B]^{3\cdot 2^{n^k}-6} \bot \to \langle \mathrm{A}\rangle \Bigg(\Big(\ell(2) \wedge \bigwedge_{a\in\Gamma} \neg a\Big) \vee \sqcup\Bigg)\Bigg) \wedge$$
$$[B]\Big(\ell\big(3 \cdot 2^{n^k} - 2\big) \to \langle \mathrm{A}\rangle\, \#\Big).$$

Finally $\psi_{window}$ enforces the window constraint: if the proposition $(d, e, f) \in \Gamma^3$ is witnessed in a subinterval (of length 2) in the subtrack of $\rho$ corresponding to the $j$-th configuration of $M$, then in the same position of (the subtrack of $\rho$ associated with) configuration $j-1$, some $(a, b, c) \in \Gamma^3$ must be there, such that $((a, b, c), (d, e, f)) \in Wnd$.

$$\psi_{window} = [B]\Bigg(\bigwedge_{i=2,\cdots,t} \bigwedge_{(d,e,f)\in\Gamma^3} \Big(\ell(3 \cdot 2^{n^k} + 3i + 1) \wedge \langle \mathrm{A}\rangle(d, e, f)$$
$$\to [B]\big(\ell(3i+1) \to \bigvee_{((a,b,c),(d,e,f))\in Wnd} \langle \mathrm{A}\rangle(a, b, c)\big)\Big)\Bigg).$$

where $t = 2^{n^k} \cdot (2^{n^k} - 4) - 1$ is encoded in binary.

All the integers which must be stored in the formula are less than $(2^{n^k})^2$, thus they need $O(n^k)$ bits to be encoded; in this way the formula can be generated in polynomial time. □

## Appendix A.6. Proof of Lemma 35

*Proof.* The proof is by induction on the complexity of $\psi$.

- $\psi = p$, with $p \in \mathcal{AP}$ ($p\ell(p) = \{p\}$). If $\mathcal{K}, \rho \models p$, then $p \in \mathcal{L}(\mathcal{K}, \rho)$ and hence $p \in \mathcal{L}(\mathcal{K}_{|p\ell(\psi)}, \rho)$. By hypothesis, it immediately follows that $p \in \mathcal{L}(\mathcal{K}'_{|p\ell(\psi)}, \rho')$, and thus $p \in \mathcal{L}(\mathcal{K}', \rho')$ and $\mathcal{K}', \rho' \models p$.

- $\psi = \neg\phi$ ($p\ell(\phi) = p\ell(\psi)$). If $\mathcal{K}, \rho \models \neg\phi$, then $\mathcal{K}, \rho \not\models \phi$. By the inductive hypothesis, $\mathcal{K}', \rho' \not\models \phi$ and thus $\mathcal{K}', \rho' \models \neg\phi$.

- $\psi = \phi_1 \wedge \phi_2$. If $\mathcal{K}, \rho \models \phi_1 \wedge \phi_2$, then in particular $\mathcal{K}, \rho \models \phi_1$. Since, by hypothesis, $\mathcal{L}(\mathcal{K}_{|p\ell(\psi)}, \rho) = \mathcal{L}(\mathcal{K}'_{|p\ell(\psi)}, \rho')$ and $reach(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{|p\ell(\psi)}, \mathrm{lst}(\rho'))$, it holds that $\mathcal{L}(\mathcal{K}_{|p\ell(\phi_1)}, \rho) = \mathcal{L}(\mathcal{K}'_{|p\ell(\phi_1)}, \rho')$ and $reach(\mathcal{K}_{|p\ell(\phi_1)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{|p\ell(\phi_1)}, \mathrm{lst}(\rho'))$, as $p\ell(\phi_1) \subseteq p\ell(\psi)$. By the inductive hypothesis, $\mathcal{K}', \rho' \models \phi_1$. The same argument works for $\phi_2$. The thesis follows.

- $\psi = \langle \mathrm{A}\rangle\, \phi$. If $\mathcal{K}, \rho \models \langle \mathrm{A}\rangle\, \phi$, there exists a track $\overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ such that $\mathrm{fst}(\overline{\rho}) = \mathrm{lst}(\rho)$ and $\mathcal{K}, \overline{\rho} \models \phi$, with $p\ell(\phi) = p\ell(\psi)$. By hypothesis, it holds that $reach(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho)) \sim reach(\mathcal{K}'_{|p\ell(\psi)}, \mathrm{lst}(\rho'))$. Hence, there exists a track $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}'}$, with $\mathrm{fst}(\overline{\rho}') = \mathrm{lst}(\rho')$, such

that $|\overline{\rho}| = |\overline{\rho}'|$ and for all $0 \leq i \leq |\overline{\rho}| - 1$, $f(\overline{\rho}(i)) = \overline{\rho}'(i)$, where $f$ is the (an) isomorphism between $reach(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho))$ and $reach(\mathcal{K}'_{|p\ell(\psi)}, \mathrm{lst}(\rho'))$. It immediately follows that $\mathcal{L}(\mathcal{K}_{|p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{|p\ell(\phi)}, \overline{\rho}')$.

We now prove that $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho})) \sim reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$. To this end, it suffices to prove that the restriction of the isomorphism $f$ to the states of $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, say $f'$, is an isomorphism between $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$ and $reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$ (note that $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$ is a subgraph of $reach(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho))$). First, it holds that $f(\mathrm{lst}(\overline{\rho})) = f'(\mathrm{lst}(\overline{\rho})) = \mathrm{lst}(\overline{\rho}')$. Next, if $w$ is any state of $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, then $f(w) = f'(w) = w'$ is a state of $reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$, as from the existence of a track from $\mathrm{lst}(\overline{\rho})$ to $w$, it follows that there is an isomorphic track (w.r.t. $f$) from $\mathrm{lst}(\overline{\rho}')$ to $w'$. Moreover, if $(w, \overline{w}) \in \delta$, then $\overline{w}$ belongs to $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, and thus $(w', f(\overline{w})) \in \delta'$ and $f(\overline{w}) = f'(\overline{w})$ belongs to $reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$. We can conclude that, for any two states $v, v'$ of $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}))$, it holds that $(v, v')$ is an edge if and only if $(f'(v), f'(v'))$ is an edge of $reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$.

By the inductive hypothesis, $\mathcal{K}', \overline{\rho}' \models \phi$ and hence $\mathcal{K}', \rho' \models \langle \mathrm{A} \rangle \phi$.

- $\psi = \langle \overline{\mathrm{B}} \rangle \phi$. If $\mathcal{K}, \rho \models \langle \overline{\mathrm{B}} \rangle \phi$, then $\mathcal{K}, \rho \cdot \overline{\rho} \models \phi$, with $p\ell(\psi) = p\ell(\phi)$, where $\rho \cdot \overline{\rho} \in \mathrm{Trk}_{\mathcal{K}}$ and $\overline{\rho}$ is either a single state or a proper track. In analogy to the previous case, let $\overline{\rho}' \in \mathrm{Trk}_{\mathcal{K}'}$ such that $|\overline{\rho}| = |\overline{\rho}'|$ and, for all $0 \leq i < |\overline{\rho}|$, $f(\overline{\rho}(i)) = \overline{\rho}'(i)$, where $f$ is the isomorphism between $reach(\mathcal{K}_{|p\ell(\psi)}, \mathrm{lst}(\rho))$ and $reach(\mathcal{K}'_{|p\ell(\psi)}, \mathrm{lst}(\rho'))$. Since $f(\mathrm{lst}(\rho)) = \mathrm{lst}(\rho')$, by definition of isomorphism, $(\mathrm{lst}(\rho), \mathrm{fst}(\overline{\rho})) \in \delta$ implies $(\mathrm{lst}(\rho'), \mathrm{fst}(\overline{\rho}')) \in \delta'$. Therefore $\mathcal{L}(\mathcal{K}_{|p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{|p\ell(\phi)}, \overline{\rho}')$ and $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho})) \sim reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\overline{\rho}'))$. Finally,

$$\mathcal{L}(\mathcal{K}_{|p\ell(\phi)}, \rho \cdot \overline{\rho}) = \mathcal{L}(\mathcal{K}_{|p\ell(\phi)}, \rho) \cap \mathcal{L}(\mathcal{K}_{|p\ell(\phi)}, \overline{\rho}) = \mathcal{L}(\mathcal{K}'_{|p\ell(\phi)}, \rho') \cap \mathcal{L}(\mathcal{K}'_{|p\ell(\phi)}, \overline{\rho}') = \mathcal{L}(\mathcal{K}'_{|p\ell(\phi)}, \rho' \cdot \overline{\rho}')$$

and $reach(\mathcal{K}_{|p\ell(\phi)}, \mathrm{lst}(\rho \cdot \overline{\rho})) \sim reach(\mathcal{K}'_{|p\ell(\phi)}, \mathrm{lst}(\rho' \cdot \overline{\rho}'))$. By the inductive hypothesis, $\mathcal{K}', \rho' \cdot \overline{\rho}' \models \phi$ and thus $\mathcal{K}', \rho' \models \langle \overline{\mathrm{B}} \rangle \phi$. □